AD-A066 374 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 20/4
INVESTIGATION OF PIPE FLOW INSTABILITY AND RESULTS FOR WAVE NUM--ETC(U)
DEC 78 M J ARNOLD

UNCLASSIFIED NL

1 OF 2
AD
A066374

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

INVESTIGATION OF PIPE FLOW INSTABILITY
AND RESULTS FOR WAVE NUMBER ZERO

by

Michael James Arnold

December 1978

Thesis Advisor: T. H. Gawain

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Investigation of Pipe Flow Instability and Results for Wave Number Zero,	(9) 5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1978
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Michael James Arnold	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	(11) 12. REPORT DATE December 1978
	13. NUMBER OF PAGES 122
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 123 p.	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pipe Flow Instability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Past research by Harrison and Johnston on the stability of pipe flow yielded only tenuous results owing to errors in setup of the problem and in formulation of the complex axis boundary conditions.

Recent advances in the formulation of these boundary conditions and application of generalized stability criteria allowed an accurate numerical solution to be made for angular wave number zero. The results show that flow for this case is characterized by certain

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

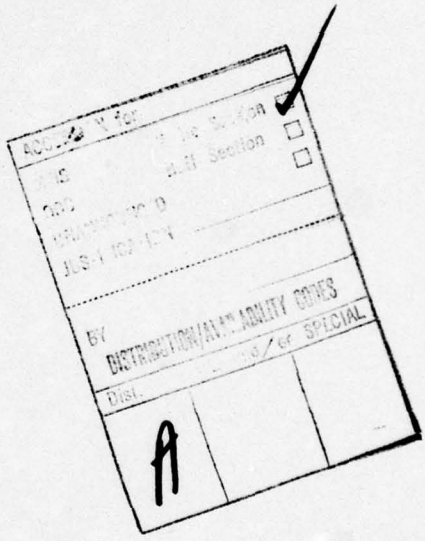
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Cont'd)

instabilities that have not been previously identified in linearized studies of this type.

A nonuniform computational mesh was developed which provided dramtic reductions in computational time on a limited basis.

Two data reduction programs were also developed to process and display data generated by the main program.



Approved for public release; distribution unlimited.

Investigation of Pipe Flow Instability and Results for Wave Number Zero

by

Michael James Arnold
Lieutenant, United States Navy
B.S., University of Idaho, 1969

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1978

Author Michael James Arnold

Approved by: T. H. Gawain
Thesis Advisor

Max F. Platzer
Chairman, Department of Aeronautics

William M. Tolles
Dean of Science and Engineering

ABSTRACT

Past research by Harrison and Johnston on the stability of pipe flow yielded only tenuous results owing to errors in setup of the problem and in formulation of the complex axis boundary conditions.

Recent advances in the formulation of these boundary conditions and application of generalized stability criteria allowed an accurate numerical solution to be made for angular wave number zero. The results show that flow for this case is characterized by certain instabilities that have not been previously identified in linearized studies of this type.

A nonuniform computational mesh was developed which provided dramatic reductions in computational time on a limited basis.

Two data reduction programs were also developed to process and display data generated by the main program.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF SYMBOLS

C	Constant in non-uniform mesh functions given by equations (C-32) and (C-40)
$D, D^2, \ldots$	Partial derivatives with respect to r.
$D^*, D^{*2}, \ldots$	Partial derivatives with respect to η.
e	Base of natural logarithms.
$\bar{e}_x, \bar{e}_r, \bar{e}_\theta$	Unit vectors along the x, r and θ axes in cylindrical coordinates.
F,G,H	Components of the velocity vector potential defined in equation (2-6).
$f_{11}, f_{22}, \ldots$	Coefficients of D^*Q, $D^{*2}Q$, ... in equations (C-9) through (C-12) as defined in equations (C-13) through (C-22).
i	$+\sqrt{-1}$, the imaginary unit. Also used as an index in Section III and Appendix D.
N	The number of interior points in the finite difference mesh of Section III.
O	Symbol denoting the phrase "of order".
Q	The component of the velocity vector potential derived from the component H by the change of variable, $H = rQ$.
R_e	Reynolds number based on mean velocity and pipe radius.
t	Time.
U	The streamwise velocity in Pipe Poiseuille Flow as defined by equation (2-11).
u,v,w	Components of the complex perturbation velocity defined in equation (D-1).
$\bar{W}$	Complex vector potential of perturbation velocity defined in equation (D-2).
x,r,θ	Cylindrical coordinates.
α	$\alpha_R + i\alpha_I$. Complex wave number of the perturbation in the x-direction.

β	in. Complex wave number of the perturbation in the θ direction, where n = 0,1,2,3, ...
δ	$1/(N+1)$. The r or η increment in the finite difference approximations of the derivatives of Q.
η	The independent variable replacing r in the nonuniform mesh of Appendix C.
γ	$\gamma_R + i\gamma_I$. Complex frequency of the perturbation.
$\overline{\Gamma}$	The vorticity transport equation expressed in abbreviated notation as defined in equation (2-7).
$\Gamma_x, \Gamma_r, \Gamma_\theta$	The components of $\overline{\Gamma}$ in cylindrical coordinates as defined in equation (2-7).
λ	Mesh offset parameter as defined in equations (C-32) and (C-40).
∇	Linear vector operator (nabla)
$\times$	Vector cross-product operator.
[ ]	Brackets enclosing a matrix.
{ }	Brackets enclosing a column vector.

I. INTRODUCTION

The problem of finding an analytical solution to the pipe flow stability problem has been pursued actively ever since the classical experiments of Osborne Reynolds [10] about 100 years ago. Up to now, however, no investigation has been able to satisfactorily predict flow instabilities, although many approaches have been taken.

Salwen and Grosch [11] studied pipe flow with various angular wave numbers and sinusoidal streamwise perturbations and concluded that it was stable for all axial and angular wave numbers. Perturbations with exponential growth in space but a purely sinusoidal time variation were researched by Garg and Rouleau [2] and those with both exponential growth in space and in time by Gill [3]. Both concluded that the flows were stable.

Because of this inability of linear theory to account for experimental fact, explanations by Davey and Drazin [1] involving finite disturbances and by Huang and Chen [5] and Leite [7] involving conditions at the pipe entrance have been offered. While these investigations have indeed shown instabilities to exist, a completely general solution to the linear problem has never been achieved.

Recently a more general theory was presented by Harrison [4] and further investigated by Johnston [6]. These two studies, however, failed to produce conclusive results due

to mathematical errors in the problem setup and inadequate formulation of the boundary conditions at the axis. Gawain [9] has subsequently formulated the axis boundary conditions in a new way which corrects the previous discrepancies and promises further advances.

For angular wave number, n, equal to zero, radical simplifications result in the governing equations (Section II), indicating that this case should be approached first. This investigation centers on that case.

Preliminary checks using the computer program of Ref. 6 revealed that, of the two eigenfunctions, G and H, which occur in this problem and which are uncoupled for n = 0, the latter appeared to be the more critical. Hence the present research was arbitrarily restricted to investigation of the stability of eigenfunction H. A similar study of the other eigenfunction, G, for n = 0 remains to be completed at some future time. Comparable calculations for other wave numbers (n = 1,2,3, ...) also remain to be accomplished in the future. Extensive and systematic calculations of this type will be essential to provide the factual basis for a comprehensive theory of pipe flow stability.

Reverting to the case at hand, eigenfunction H for wave number n = 0, we note that the program of Ref. 6 was rewritten for this case, incorporating the newly formulated boundary conditions of Ref. 9. In addition, a new, generalized stability criteria was adopted. Moreover, a new technique was introduced which allows the use of nonuniform meshes to reduce computational time.

Lastly, two data reduction programs were written to process data produced by the main investigative program.

II. THE VORTICITY TRANSPORT EQUATION

Although a complete treatment of this subject is contained in Appendix A of Ref. 4 and further addressed in Ref. 6 and Ref. 9, it is felt that a brief overview is still required here to maintain continuity with previously referenced works. This discussion is an abbreviated version of Section II of Ref. 6.

Laminar flow of an incompressible fluid of constant viscosity is governed by the Navier-Stokes equation and the continuity equation. Taking the curl ($\nabla\times$) of the Navier-Stokes equation and introducing a perturbation velocity ($\bar{v}$) and vorticity ($\bar{\omega}$) gives the vorticity transport equation which is equation (A-10) of Appendix A, Ref. 4.

Expressing this equation in terms of the complex velocity vector potential, $\bar{W}$, gives

$$W(x,r,\theta,t) = (\bar{e}_x F(r) + \bar{e}_r G(r) + \bar{e}_\theta H(r))e^X \qquad (2\text{-}1)$$

where

$$X = \alpha x + \beta\theta + \gamma t \qquad (2\text{-}2)$$

and

$$\bar{v} = \nabla \times \bar{W} \qquad (2\text{-}3)$$

$$\bar{\omega} = \nabla \times \bar{v} \, . \tag{2-4}$$

It should also be noted that, as shown in part one of Appendix G in Ref. 4, α and γ are complex while β is a purely imaginary quantity defined by

$$\beta = in \qquad n = 0,1,2, \ldots \tag{2-5}$$

When expressed in the form of equation (2-1), the vorticity transport equation becomes three simultaneous fourth-order differential equations of the form

$$[M_4] \begin{Bmatrix} D^4F \\ D^4G \\ D^4H \end{Bmatrix} + [M_3] \begin{Bmatrix} D^3F \\ D^3G \\ D^3H \end{Bmatrix} + [M_2] \begin{Bmatrix} D^2F \\ D^2G \\ D^2H \end{Bmatrix}$$

$$+ [M_1] \begin{Bmatrix} DF \\ DG \\ DH \end{Bmatrix} + [M_0] \begin{Bmatrix} F \\ G \\ H \end{Bmatrix} - \gamma([N_2] \begin{Bmatrix} D^2F \\ D^2G \\ D^2H \end{Bmatrix}$$

$$+ [N_1] \begin{Bmatrix} DF \\ DG \\ DH \end{Bmatrix} + [N_0] \begin{Bmatrix} F \\ G \\ H \end{Bmatrix}) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{2-6}$$

Equations (2-5) may be further expressed in the abbreviated form

$$\overline{\Gamma} = \begin{Bmatrix} \overline{\Gamma}_x \\ \overline{\Gamma}_r \\ \overline{\Gamma}_\theta \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{2-7}$$

where $\overline{\Gamma}$ appears to be a set of three coupled equations in the components of $\overline{W}$. As given in Appendix B of Ref. 4, equations (2-7) actually represent only two independent conditions and by an appropriate linear combination of Γ_x and Γ_θ, equations (2-6) can be expressed as a set of two equations in three unknowns. The appropriate linear combination is given in Appendix B of Ref. 4 and yields the set of equations

$$\begin{aligned} \Gamma_r &= 0 \\ -\frac{in}{r}\Gamma_x + \alpha\Gamma_\theta &= 0 \ . \end{aligned} \tag{2-8}$$

Except for the case where n is equal to zero, equations (2-8) do not uncouple. The linear combination given by the second of equations (2-8) does, however, reduce the highest order derivative of G(r) in equations (2-6) to second order. Appendix C of Ref. 4 illustrates the redundancy of the three components of $\overline{W}$, allowing one of these components to be arbitrarily set to zero for all r. The maximum benefits of equations (2-8) are obtained if

$$F(r) = 0 \tag{2-9}$$

Incorporating equations (2-8) and (2-9) into equations (2-6) results in the form

$$[M_4'] \begin{Bmatrix} D^4G \\ D^4H \end{Bmatrix} + [M_3'] \begin{Bmatrix} D^3G \\ D^3H \end{Bmatrix} + [M_2'] \begin{Bmatrix} D^2G \\ D^2H \end{Bmatrix}$$
$$+ [M_1'] \begin{Bmatrix} DG \\ DH \end{Bmatrix} + [M_o'] \begin{Bmatrix} G \\ H \end{Bmatrix} - \gamma([N_2'] \begin{Bmatrix} D^2G \\ D^2H \end{Bmatrix}$$
$$+ [N_1'] \begin{Bmatrix} DG \\ DH \end{Bmatrix} + [N_o'] \begin{Bmatrix} G \\ H \end{Bmatrix}) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad (2\text{-}10)$$

where the coefficient matrices are given by equations (2-10) through (2-17) of Ref. 6. It is appropriate to note that these same coefficient matrices appear in Ref. 9, equations (A1) through (A9), in a slightly different form resulting from the substitutions

$$U = 2(1 - r^2) \qquad (2\text{-}11)$$

$$t = \alpha^2 + \frac{\beta^2}{r^2} \quad \text{and} \qquad (2\text{-}12)$$

$$T = \alpha U - \frac{1}{R_e}(\alpha^2 + \frac{\beta^2}{r^2}) \ . \qquad (2\text{-}13)$$

As discussed in the previous section, the case where

$$\beta = in \ , \quad n = 0 \qquad (2\text{-}14)$$

leads to great simplifications in equations (2-10), (2-12) and (2-13). In particular, equations (2-10) uncouple and allow an independent investigation of either H or G. As a result of the findings discussed in Section I, it was decided to explore the function H only. This reduced equation (2-10) to that of equation (A-6) of Appendix A, which is a linear, homogeneous fourth order differential equation in H(r).

III. NUMERICAL METHODS

Substituting the change of variable $H = rQ$ as given in equation (A-1) and the coefficients defined in equations (A-11) through (A-18) into the vorticity transport relation, equation (A-6), gives the expression

$$M_4 D^4 Q + M_3 D^3 Q + M_2 D^2 Q + M_1 DQ + M_0 Q - \gamma [N_2 D^2 Q + N_1 DQ + N_0 Q] = 0 , \qquad (3\text{-}1)$$

which is a homogeneous fourth order differential equation in $Q(r)$. The boundary conditions for this case are derived in detail in Ref. 9 as

$$\begin{aligned} Q(1) &= 0 \\ DQ(1) &= 0 \\ DQ(0) &= 0 \\ D^3 Q(0) &= 0 . \end{aligned} \qquad (3\text{-}2)$$

The boundary finite difference equations derived in Appendix B from equations (3-2), along with the standard central difference equations given in Ref. 6, allow the function $Q(r)$ to be approximated by a finite number of discrete unknowns. As shown by Figure 3-1 below, the non-dimensionalized radius of the pipe is divided into a one-dimensional

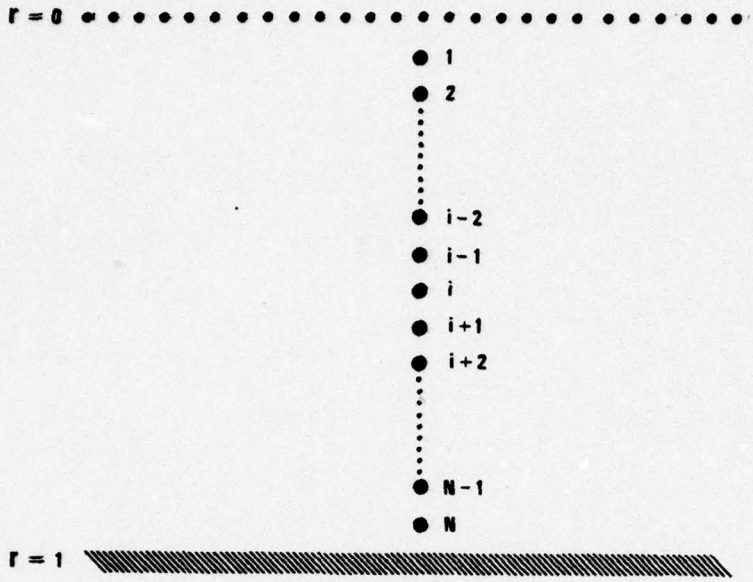


Figure 3-1 Finite Difference Mesh

computational mesh consisting of N interior points, N+1 intervals, and N+2 total points, including the boundary points at r = 1 and r = 0. As will be discussed later, the spacing between these points may or may not be uniform. For the uniform case, the spacing is defined by

$$\delta = 1/(N+1) \ . \qquad (3\text{-}3)$$

For the nonuniform case, a change of independent variable is performed. The spacing of the new independent variable, η, is still given by equation (3-3).

With a nonuniform mesh, the points shown in Figure 3-1 will be concentrated near the axis or near the wall according

to the type of offset specified. These effects are discussed in detail in Section IV.

Substitution of the finite difference equations of Appendix B into equation (3-1) results in a set of N, linear, algebraic difference equations in terms of the unknown value of Q at each of the N interior points of the computational mesh. Since each of these equations is of the form of a linear combination of the ith, central, point and the two, three or four adjacent points (depending on the order of the derivative being approximated), this system of equations consists of a coefficient array multiplying a vector containing the unknown value of the function Q at each of the N interior points. This technique allows the problem to be converted into an eigenvalue problem of the form

$$[X]\ \{Q\} - \gamma[Y]\ \{Q\} = 0 \qquad (3\text{-}4)$$

with the basic composition of the arrays [X] and [Y] and the vector {Q} as illustrated in Figure 3-2 below.

It should be noted at this point that Figure 3-2 differs somewhat from the normal finite difference banded matrix in the first two rows and last row because of the method of deriving the finite difference approximations at the boundaries. Additionally, the order of the N unknowns has been reversed from that of Ref. 6. This was done to conform to standard matrix notation.

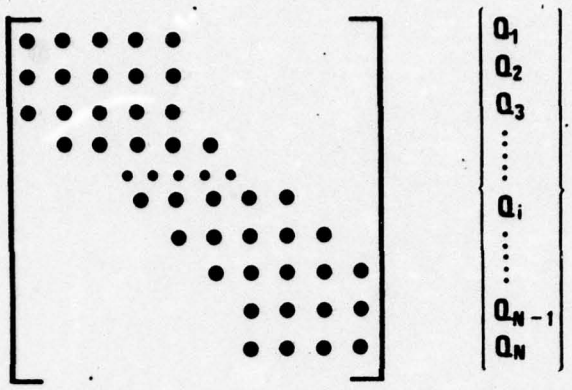


Figure 3-2 Basic Composition of Coefficient Arrays and Vector of Unknowns

This array is established by the subroutine MSET2 in conjunction with the subroutine MSET1 and function subprograms CQM1E1 and CQM2E1, which compute the numerical value for each element in the array. Subroutine MSET1 provides the coefficients given by equations (A-11) through (A-18) of Appendix A or by equations (C-24) through (C-31) if a nonuniform mesh is specified. Function CQM1E1 then computes the values for each of the elements of array [X] in equation (3-4) using the coefficients passed from subroutine MSET1 in vector CQM1. Function subprogram CQM2E1 performs the same function for matrix [Y] in equation (3-4) using the coefficients passed in vector CQM2.

The solution of the eigenvalue problem as formulated to this point is carried out by the controlling subroutine of program PIPE0, subroutine STAB, by the following steps:

1) Subroutine MSET2 is called twice to set up the coefficient matrices [X] and [Y] of equation (3-4).

2) Subroutine CDMTIN is then called to invert matrix [Y], the second coefficient array in equation (3-4). CDMTIN was obtained from the IBM Library routine CMTRIN by modifying it to accept double precision arrays.

3) Both coefficient arrays, [X] and [Y], are then pre-multiplied by $[Y]^{-1}$. Since multiplication of an array by its inverse invariably results in the identity matrix, [I], only the product $[Y]^{-1}[X]$ is computed using subroutine MULM. This converts the eigenvalue problem of equation (3-4) to the more conventional form

$$([Z] - \gamma[I])\{Q\} = 0 \qquad (3\text{-}5)$$

where

$$[Z] = [Y]^{-1}[X] \qquad (3\text{-}6)$$

4) Since all programs currently available for solving equations (3-5) require that the real and imaginary parts of the elements of [Z] be presented in separate arrays, subroutine DSPLIT is called to accomplish this.

5) The eigenvalues and eigenvectors of equations (3-5) are computed using subroutines EBALAC, EHESSC, ELRH2C and EBBCKC which are available through the International

Math and Statistics Library. Subroutine EBALAC balances matrix [Z] by equalizing the exponents of all terms. The details of this transformation are retained for later use. The balanced matrix is then passed to subroutine EHESSC where it is reduced into the complex upper Hessenberg form. Subroutine ELRH2C then solves for the eigenvalues and eigenvectors. To transform the eigenvectors back into the original unbalanced form, EBBCKC is finally called using information passed from subroutine EBALAC.

For each solution, subroutine STAB determines the least stable eigenvalue (largest algebraic value) and then writes the values of N, R_e, α_R, α_I, λ, γ_{RL}, γ_{IL} and KSET to file FT02F001. The eigenvector corresponding to the least stable eigenvalue is also written to FILE FT02F001 when MODENO is set equal to one.

Control of subroutine STAB is accomplished by the main program, PIPE0. This program is a time-sharing (CP/CMS) program. Modes one and three compute the stability of the flow for a given set of input conditions. Mode one writes the least stable eigenvector to FILE FT02F001 while this output is inhibited when MODENO is set equal to three. To generate data for program EIGFCN, program PIPE0 must be run with MODENO equal to one.

Mode two operation generates a grid of stability values (stability map) based on parameters read in from FILE FT01F001. Due to the long run time in this mode, only small meshes can be generated under CP/CMS. Longer runs must be accomplished under batch, with changes to the program as specified in the comments section. Data is output to file FT03F001 when MODENO is equal to two and is compatible with program STBCONT.

The plotting programs EIGFCN and STBCONT were used to process the data generated by program PIPE0 in modes one and three, respectively. Program EIGFCN generates normalized plots of the perturbation velocity, u, as a function of radius, r. The perturbation velocities generated in accordance with Appendix D were normalized in two steps. First the perturbation velocity of largest magnitude was determined. Letting this velocity be termed u_C, a normalizing constant producing unit magnitude and zero phase angle in u_C was found in the following manner:

If

$$u_C = u_{RC} + iu_{iC} , \qquad (3-7)$$

then

$$Cu_C = 1 + i(0) \qquad (3-8)$$

where C is the normalizing constant. Thus,

$$C = \frac{1}{u_{RC} + iu_{iC}} = \frac{u_{RC} - u_{iC}}{(u_{RC}^2 - u_{iC}^2)} \qquad (3\text{-}9)$$

$$= \frac{\bar{u}_C}{|u_C|^2} \qquad (3\text{-}10)$$

where $\bar{u}_C$ is the complex conjugate of u_C.

The nondimensionalized radius values were taken directly from the data cards for uniform meshes or computed from equations (C-32) or (C-40) in the case of a nonuniform mesh.

Program STBCONT plots the stability contours against α_R and α_I. The stability map generated by program PIPE0 is searched columnwise and rowwise for sign changes for each of the three stability criteria discussed in Section V and Ref. 9. The points are then plotted, producing contours of incipient, critical and fully developed instability and areas that denote stable flow and subcritical, supercritical and hypercritical instability.

Both programs, EIGFCN and STBCONT, utilize the NPS VERSATEC plotter, certain built-in VERSATEC subroutines, and subroutine PLOTG. These routines are only accessable when running under FORTCLGW.

IV. RESULTS

A. STABILITY

Since an understanding of the term stability is necessary to interpret the results of this investigation, a brief discussion is presented here. A complete discussion of the generalized criteria of stability is given by Gawain [9].

The characteristics of the flow for the case n = 0 are set by the parameters R_e and α. For fixed values of these parameters, the solution of equations (3-5) is a set of N eigenvalues, γ, and their corresponding eigenvectors, Q. As can readily be seen from equation (2-1), the value of the real part of the complex eigenvalue γ will determine the growth or decay rate in time of the perturbation. Since positive values of the real part of γ represent an exponential growth rate in time, the most important γ is the one having the largest algebraic value for its real part. This root is termed the least stable root and will be represented by the symbol γ_{RL}. As the stability represented by γ_{RL} is that seen by a fixed observer, it is not the most general criterion. As derived in Ref. 9, a more appropriate stability criterion is that based on an axis system moving at the average volumetric velocity of the flow. This criteria is termed γ^*_{RL} and is defined by Ref. 9 as

$$\gamma_{RL}^{*} = \gamma_{RL} + \alpha_R \, . \qquad (4\text{-}1)$$

For this and subsequent discussions, the subscript will be dropped and γ^* will refer to the quantity defined by equation (4-1). Three stability cases arise from this equation. The first is termed incipient instability and is defined by

$$\gamma^* = -|\alpha_R| \, . \qquad (4\text{-}2)$$

The second case, termed critical instability, is given by

$$\gamma^* = 0 \qquad (4\text{-}3)$$

and, lastly, the case termed fully developed instability is said to exist when

$$\gamma^* = + |\alpha_R| \, . \qquad (4\text{-}4)$$

The transition from stable flow to fully developed instability is progressive and several distinct stages are given in Ref. 9 to describe this transition. The region from incipient to critical instability is termed subcritical instability, that from critical instability to fully developed instability is called supercritical instability while that beyond fully developed instability is termed hypercritical instability.

B. PERTURBATION VELOCITY PLOTS

Initial investigation of the function Q was centered around plotting its appearance in the region of interest. A Reynolds number of 1150 (2300 based on diameter) was chosen as this value is generally accepted as the nominal value for transition to turbulent flow. The value of α was set at -0.5 + i 10.0 for the major part of the investigation as preliminary checks revealed that supercritical instabilities were present for this value. A secondary Reynolds number of 4000 was chosen to show trends.

The quantity chosen as the most realistic and representative of the eigenfunction Q is the axial perturbation velocity, u. This quantity was derived from the elements of the least stable eigenvector as outlined in Appendix D. Initially, R_e and α_I were held fixed and α_R was varied over a range of positive and negative values. For values of α_R below about two, the normalized perturbation velocity was found to have all activity near the axis with a decay essentially to zero by r = 0.3. A typical plot of u versus r for an α_R in this range is shown in Figure 4-1. When α_R was made sufficiently positive, the plot changed significantly in both appearance and region of activity. Figure 4-2 shows a plot of u for α_R = 2.5. The activity can now be seen to be concentrated near the wall, with most of the activity occurring at r values greater than 0.7.

Although no particular relationship between the nature of u and the stability of the flow was evident or expected,

the plots were nevertheless valuable as indicators for various parameters involved in the investigation.

First, as can be seen by the differences in Figures 4-1 and 4-2, the plots were ideal indicators of changes in the nature of the function Q. Secondly, the adequacy of the mesh could be directly observed by noting the number of points defining the curves in regions of high activity. Figures 4-3, 4-4 and 4-5 show the same conditions as Figure 4-1 but with decreasing number of mesh points, N. Lastly, the effects of nonuniform meshes could be observed as will be discussed later in this section.

C. STABILITY CONTOUR PLOTS

The principal results of this investigation are shown in Figures 4-6 and 4-7. Although these two figures pertain to only a limited portion of the complex α plane, they do represent a significant advance in the investigation of pipe flow stability. As can be seen in these figures, the flow is characterized by regions of differing stability, ranging from stable through supercritical instability. Note that these two figures correspond to Reynolds numbers of 1150 and 4000, respectively. This is a result that has not, to this writer's knowledge, been heretofore achieved by a linearized analysis of fully developed pipe flow. The figures also show that, as has been born out by previous investigations, flow for purely sinusoidal oscillations ($\alpha_R = 0$) is stable. Additionally, a comparison of Figures 4-6

and 4-7 shows the effect of Reynolds number on the flow stability. It is clear from this comparison that an increase in Reynolds number reduces the size of the stable regions in the complex α plane; in other words, stability decreases with increasing Reynolds number. This trend agrees with our general experience pertaining to fluid flow. Lastly, the effect of the real and imaginary parts of the wave number α can readily be seen. For α_R, increasingly negative values produce successively greater levels of instability. While a contour plot was not produced for positive values of α_R, point checks of stability in this region suggest that somewhat similar contours exist in the right half-plane also. For α_I, increasing values produce increasing stability. This effect is also more pronounced at the lower Reynolds number.

D. NONUNIFORM MESH EFFECTS

One of the difficulties in this investigation was the relatively long computing time required to obtain an accurate solution, especially when operating under CP/CMS (time-sharing). The major factor controlling computing time was the number of interior mesh points, N. As an example, an increase in N of 50 percent resulted in a fourfold increase in computing time. Therefore, the desired objectives of rapidity and accuracy were in direct conflict. Additionally, follow-on investigations for values of angular wave number n other than zero involve matrices twice the order required for this case because of the coupling of equations (2-8).

For these reasons, a nonuniform mesh was developed to obtain increased accuracy at lower values of N. The nature of the velocities as seen in Figures 4-1 and 4-2 shows that a high degree of resolution in the computational mesh is only required in the vicinity of the axis (α_R less than about 2) or the wall (α_R greater than about 2). It was therefore theoretically possible to redistribute the points at moderate values of N to attain resolutions equivalent to much finer (and more time-consuming) uniform meshes.

As can be seen from Figures 4-8 and 4-9, the value of γ^* varies with the number of mesh points, N. Theoretically, each of these curves would approach some limiting value if N were increased without bound, and it is this theoretical limit that represents the required solution. In practice, it is adequate to approximate the unknown limit by a point that lies on the relatively flat portion of the curve at a value of N which is practically attainable and which does not involve a prohibitively long computing time. It has been found in this investigation that N = 79 fulfills these conditions.

The conversion to a nonuniform mesh involved a change of independent variable and the introduction of an analytical function to control the distribution of the mesh points. The details of these steps are given in Appendix C. By varying the mesh offset parameter, λ, it was possible to vary γ^* over a wide range. To determine when the high

accuracy solution (N = 79) and the nonuniform solutions were approximately equal, γ^* was plotted versus λ for fixed values of R_e, α and N with the value of γ^* for N = 79 as a reference. Figure 4-10 shows a plot of this type for N = 31. The appropriate value of λ can be seen to be approximately 1.1. Figure 4-11 is the perturbation velocity plot of the solution for N = 31 and λ = 1.1 for the same R_e and α as Figure 4-1. Note that the γ^* values are equal for these two figures. While the resolution of Figure 4-11 is not quite as fine as that of Figure 4-1, a comparison of Figure 4-11 with Figure 4-5 makes the improved resolution obvious. Figures 4-2 and 4-12 are similar to Figures 4-1 and 4-11 except that a wall offset was used. Note that for this case λ = 1.2, which points to a drawback of the nonuniform mesh, that of dependence on input conditions. While a check of λ dependence on α was not made, it most probably exists. There is also, however, the possibility that for small regions of the complex α plane, the variations in λ are small enough to allow an average value of λ to be nearly optimum for the entire region. While not used for the main results of this study, the method as developed here may well prove to be of maximum utility in follow-on investigations of higher angular wave numbers.

E. NUMERICAL ACCURACY

To ensure that the solutions presented here were of sufficient accuracy, two separate checks were made. The

first, γ^* dependence on N, is the most commonly used criterion.

For a solution to be accurate, it should be virtually independent of mesh fineness, that is, of N. The required magnitude of N for an accurate solution was found by plotting γ^* against N. Figures 4-8 and 4-9 both show that the solution is well converged for N = 79 at Reynolds numbers of 1150 and 4000, as γ^* changes by only .001 to .003 from N = 31 to N = 79 for both values of Reynolds number.

The second verification of the solution, so obvious that it is sometimes overlooked, involves simply substituting the numerical solution (least stable eigenvector) into the governing equation to ensure that it is indeed being satisfied. A short program was independently written to check the finite difference representations of equation (3-1) at the first and last interior stations and at a mid-radius station. Initial checks of numerical solutions yielded unsatisfactory results and led to the discovery of various programming errors. In particular, it was discovered that four double precision constants in the finite difference approximations were lacking the required "D0" exponent. Elimination of these seemingly trivial errors resulted in a surprising four order-of-magnitude improvement in the accuracy of the solution, with the left side of equation (3-1) improving from order 10^{-4} to order 10^{-8}.

It is instructive to note at this point that the order of magnitude of the left side of equation (3-1) is not the

true measure of its satisfaction. A more correct procedure is to compare this value with the largest term in the equation. When examined from this viewpoint, the relative error for solutions at $R_e = 1150$ and $R_e = 4000$ are found to be of order 10^{-11} to 10^{-12}, a very satisfactory result.

Therefore, by these results, the solutions presented here are both virtually independent of N and satisfy the governing differential equation to a high degree. The efforts expended to reach these conclusions were well worth the result and also point out that attention to detail is fundamental to accurate numerical results.

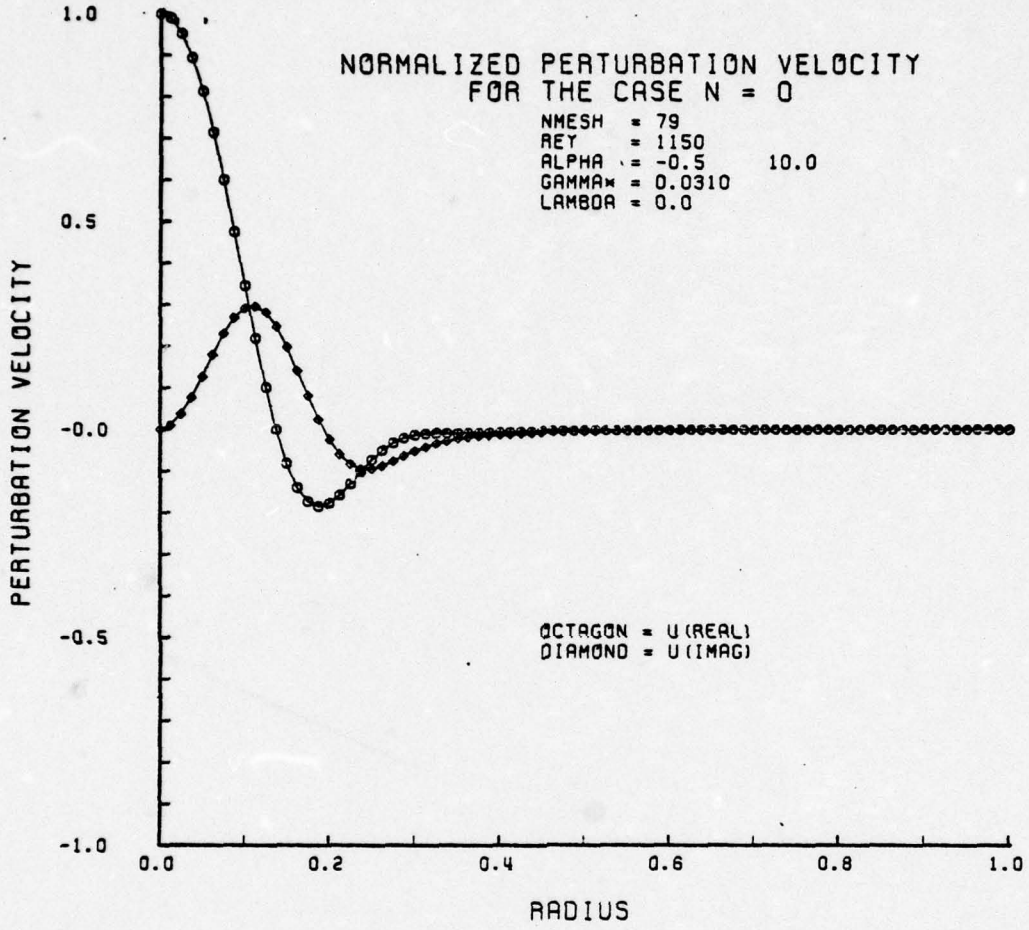

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 79
REY = 1150
ALPHA = -0.5 10.0
GAMMA = 0.0310
LAMBDA = 0.0
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-1

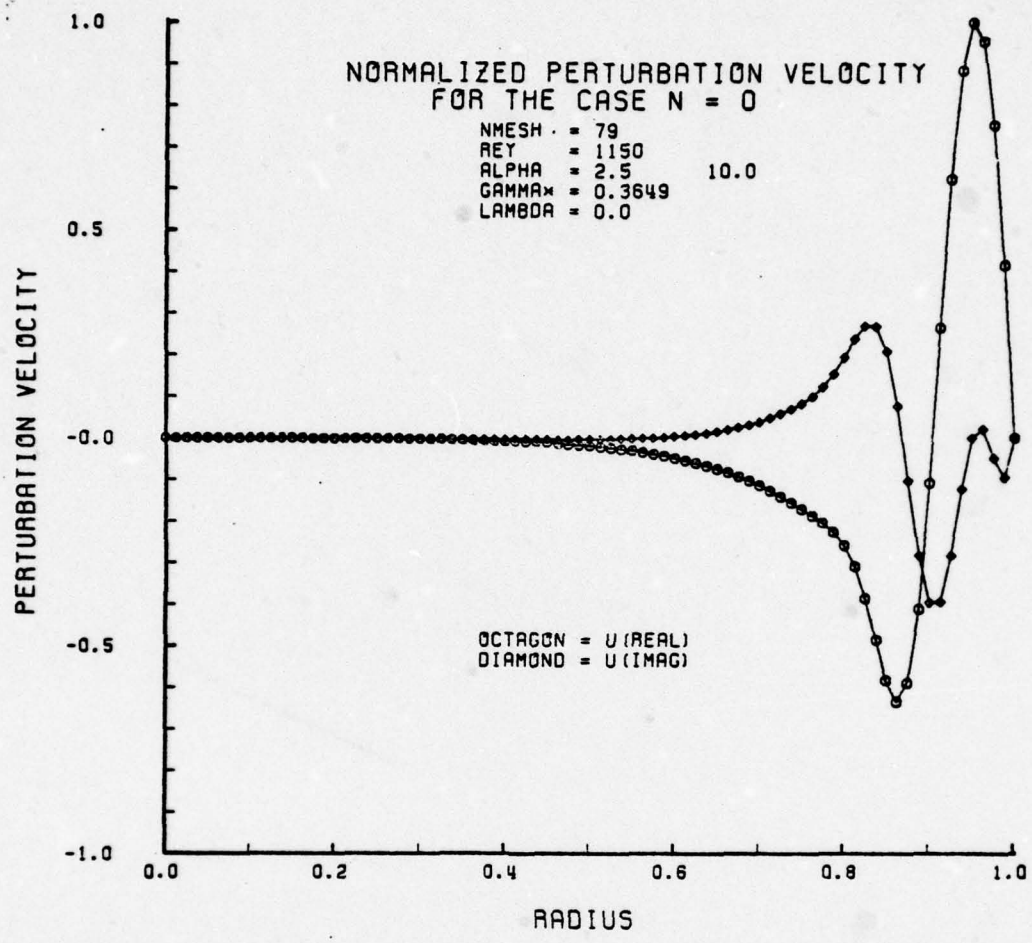

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 79
REY = 1150
ALPHA = 2.5 10.0
GAMMAx = 0.3649
LAMBDA = 0.0
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-2

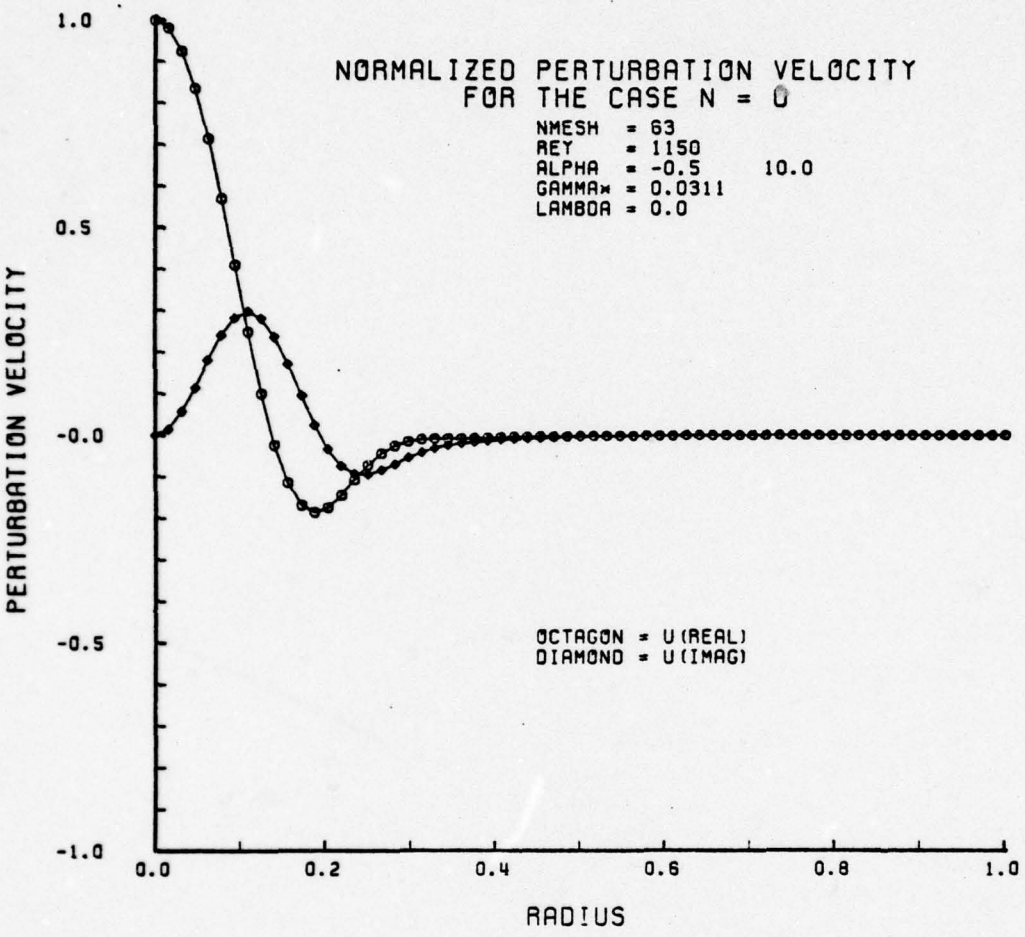

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 63
REY = 1150
ALPHA = -0.5 10.0
GAMMAx = 0.0311
LAMBDA = 0.0
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0
RADIUS

FIGURE 4-3

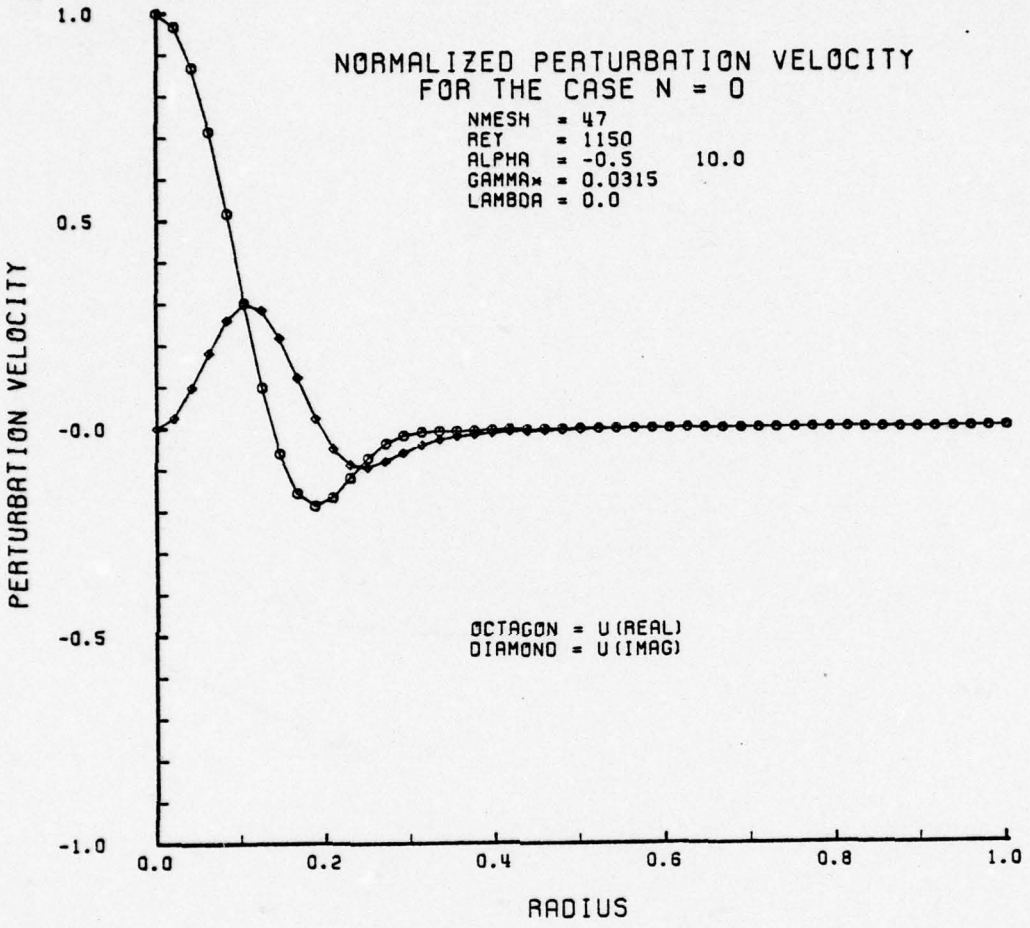

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 47
REY = 1150
ALPHA = -0.5 10.0
GAMMA* = 0.0315
LAMBDA = 0.0
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-4

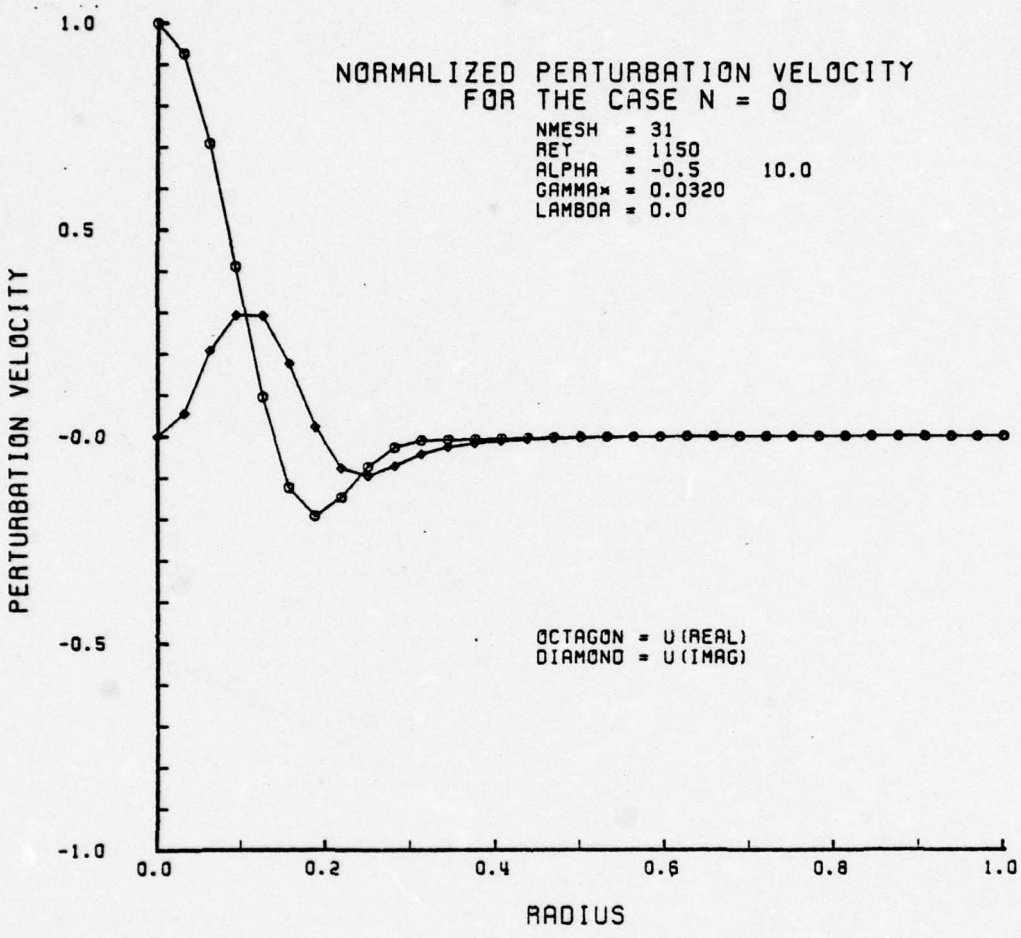

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 31
REY = 1150
ALPHA = -0.5 10.0
GAMMAx = 0.0320
LAMBDA = 0.0
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-5

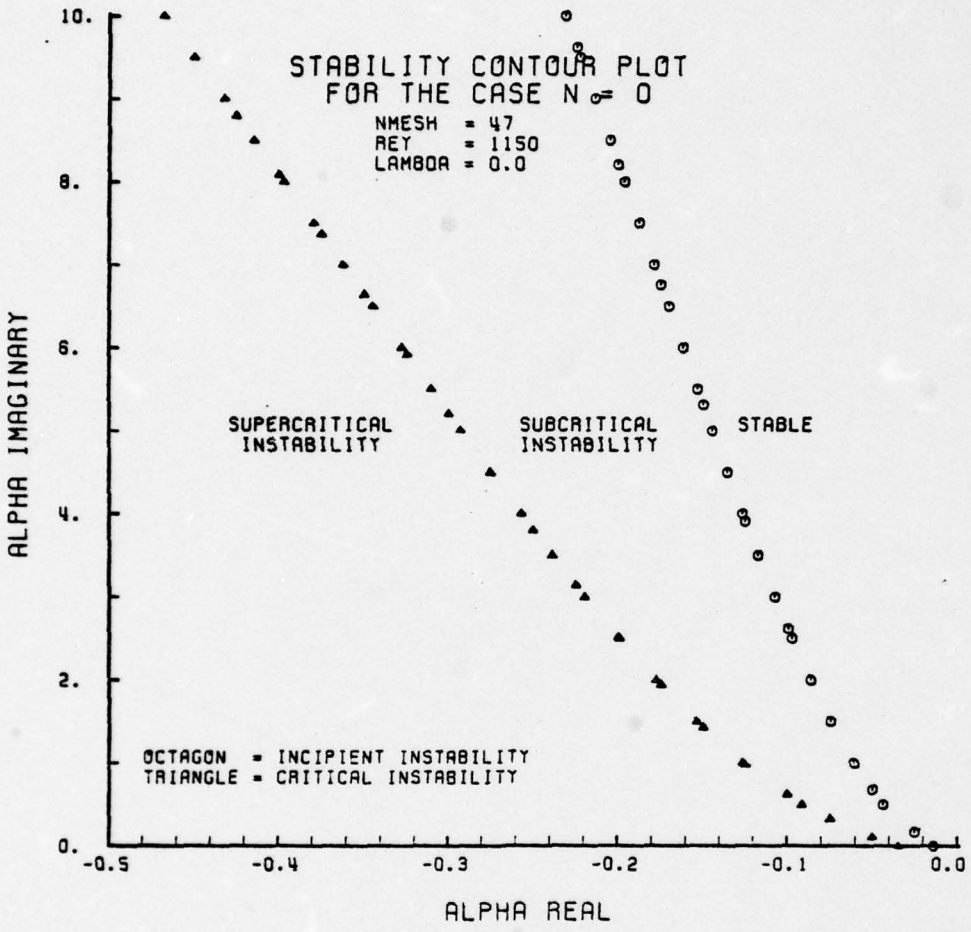

STABILITY CONTOUR PLOT
FOR THE CASE N = 0
NMESH = 47
REY = 1150
LAMBDA = 0.0
SUPERCRITICAL INSTABILITY
SUBCRITICAL INSTABILITY
STABLE
OCTAGON = INCIPIENT INSTABILITY
TRIANGLE = CRITICAL INSTABILITY
ALPHA IMAGINARY
ALPHA REAL
10.
8.
6.
4.
2.
0.
-0.5
-0.4
-0.3
-0.2
-0.1
0.0

FIGURE 4-6

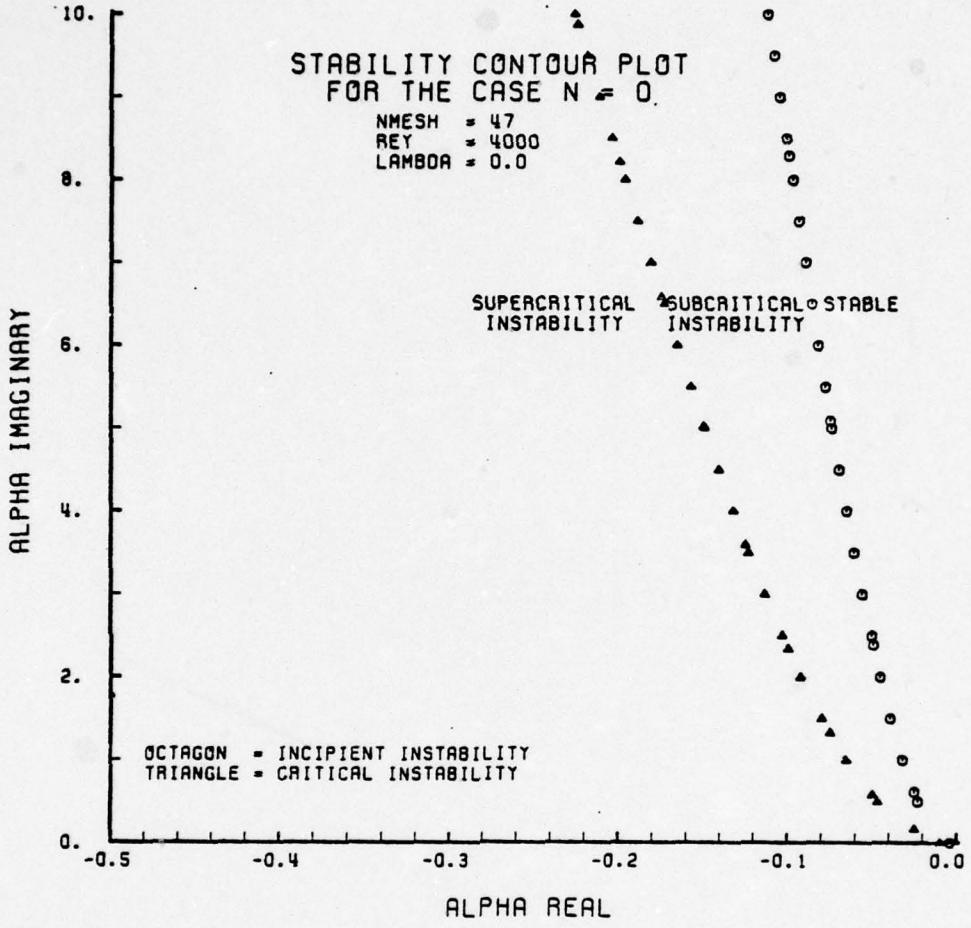

STABILITY CONTOUR PLOT
FOR THE CASE N = 0.
NMESH = 47
REY = 4000
LAMBDA = 0.0
SUPERCRITICAL INSTABILITY
SUBCRITICAL INSTABILITY
STABLE
OCTAGON = INCIPIENT INSTABILITY
TRIANGLE = CRITICAL INSTABILITY
ALPHA IMAGINARY
ALPHA REAL
10.
8.
6.
4.
2.
0.
-0.5
-0.4
-0.3
-0.2
-0.1
0.0

FIGURE 4-7

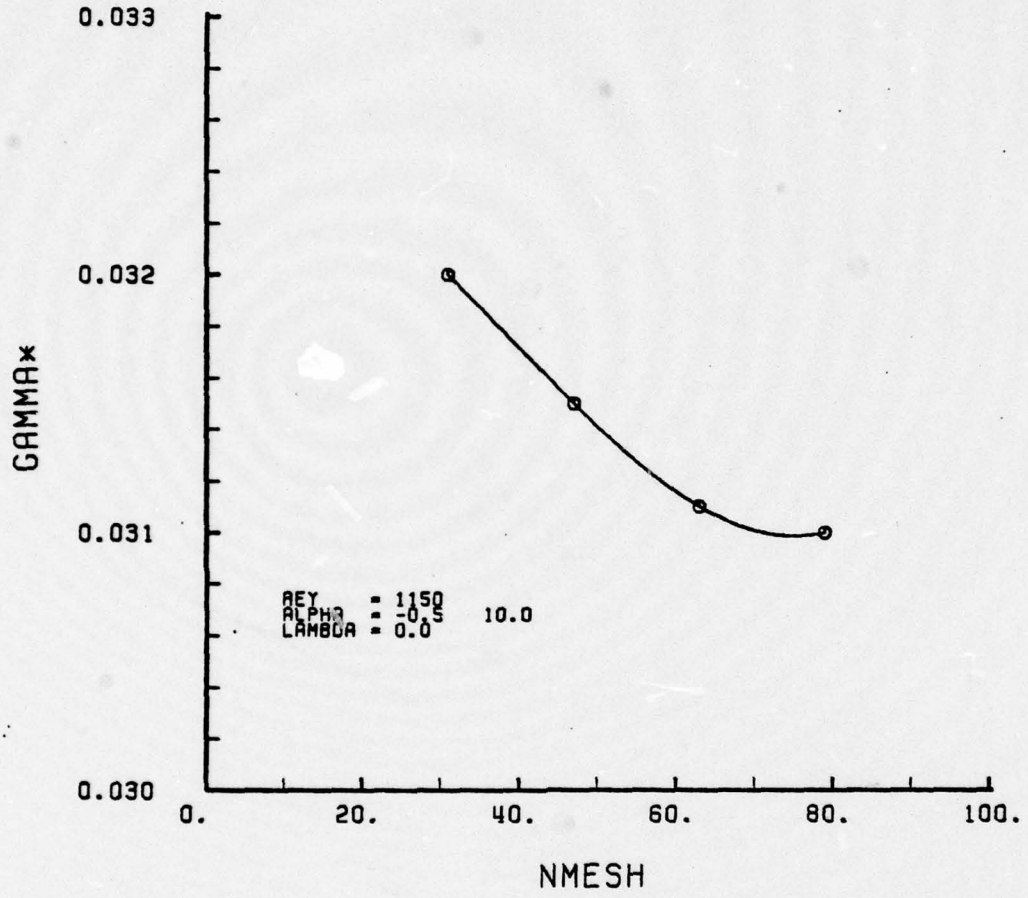


FIGURE 4-8. γ^* Versus Number of Mesh Points, N.

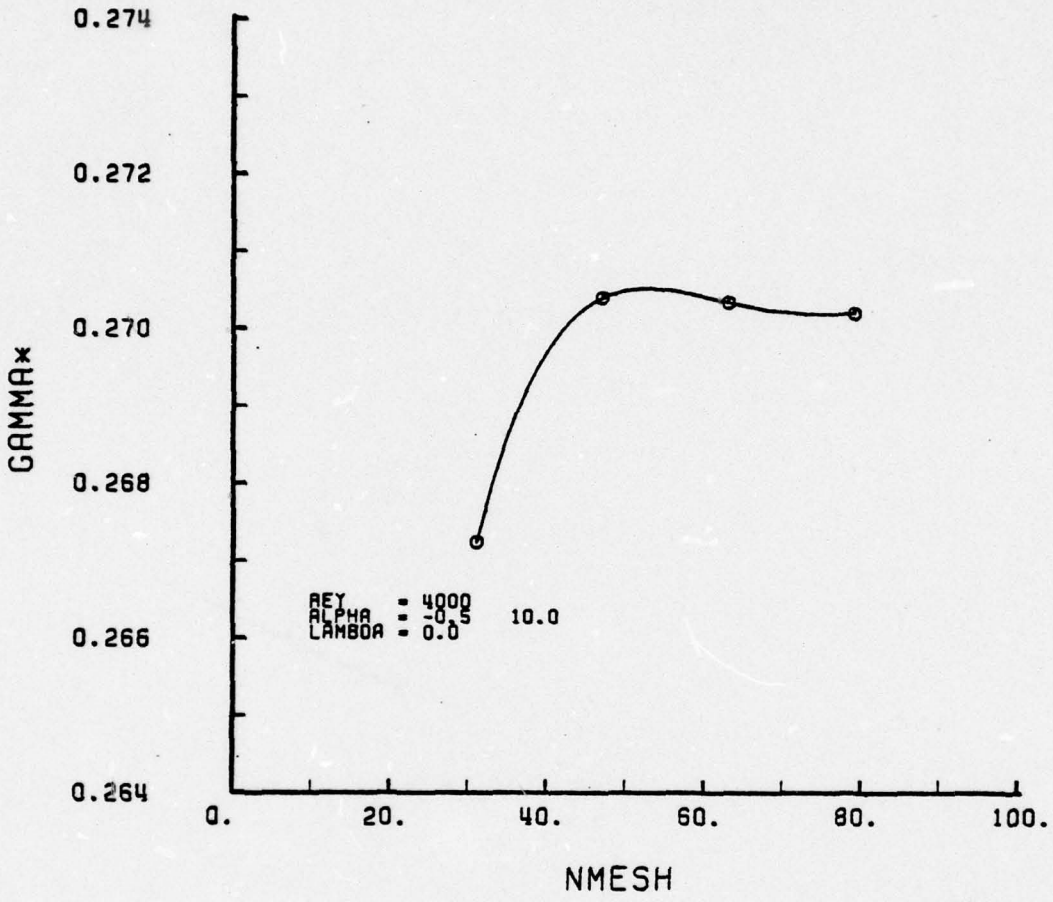


FIGURE 4-9. γ^* Versus Number of Mesh Points, N.

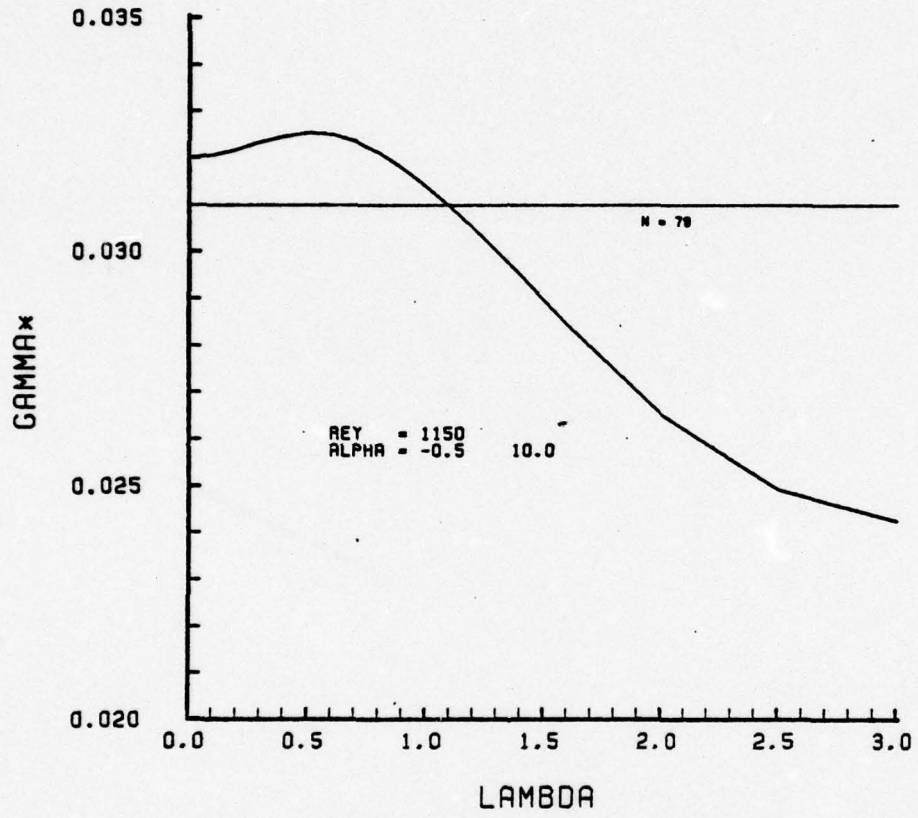


FIGURE 4-10. γ^* Versus Mesh Parameter, Lambda

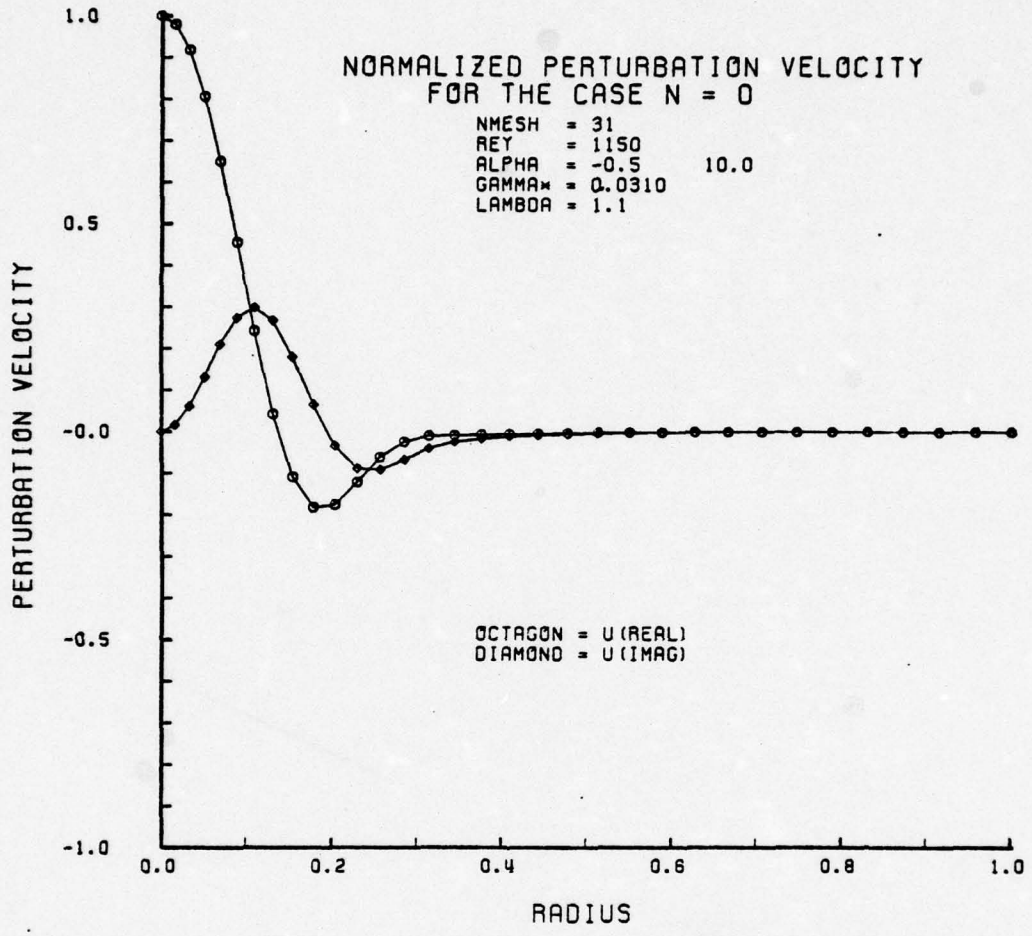

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 31
REY = 1150
ALPHA = -0.5 10.0
GAMMA* = 0.0310
LAMBDA = 1.1
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-11

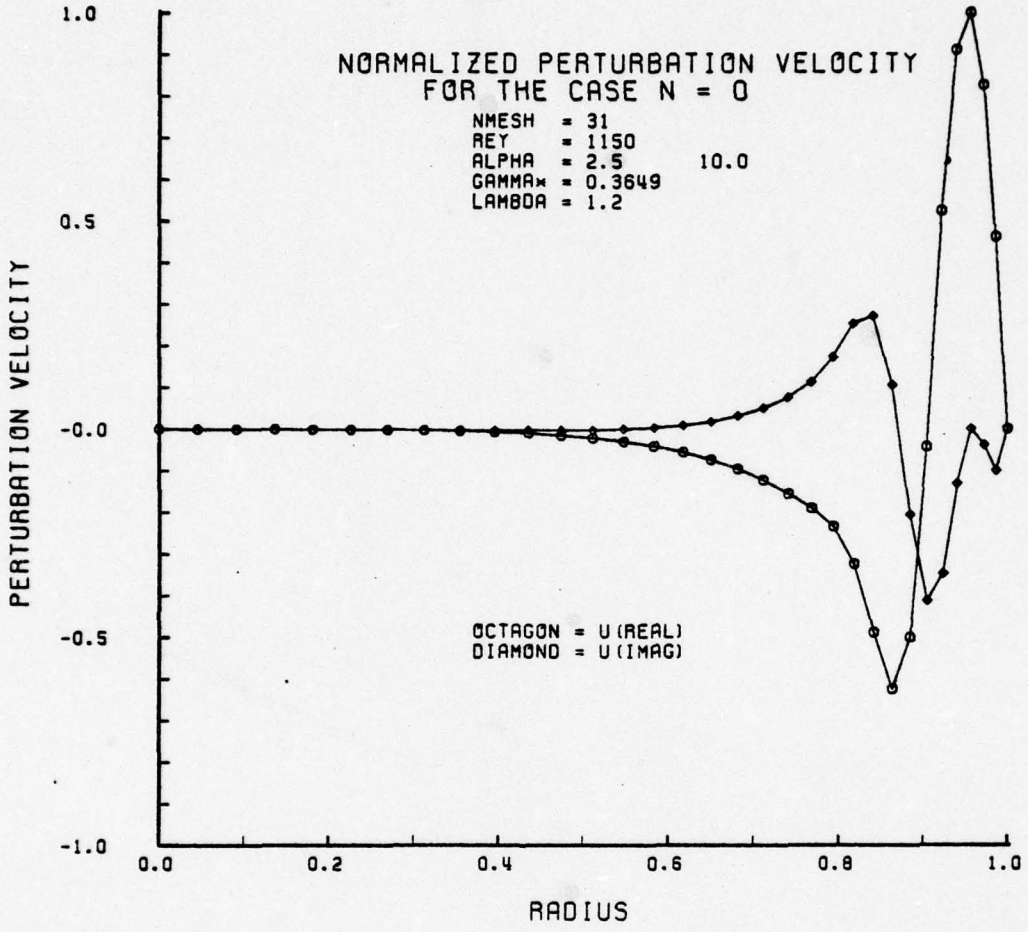

NORMALIZED PERTURBATION VELOCITY
FOR THE CASE N = 0
NMESH = 31
REY = 1150
ALPHA = 2.5 10.0
GAMMAx = 0.3649
LAMBDA = 1.2
OCTAGON = U(REAL)
DIAMOND = U(IMAG)
PERTURBATION VELOCITY
RADIUS
1.0
0.5
-0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

FIGURE 4-12

V. CONCLUSIONS AND RECOMMENDATIONS

The implementation of the newly developed boundary conditions of Gawain [9] has permitted a stable, numerical solution to the linearized vorticity transport equation. The results of the numerical solution are presented in Section IV and show that the stability of pipe Poiseuille flow is governed by the three parameters, α_R, α_I and R_e. In particular, both positive and negative values of α_R, that is, streamwise growth and decay in space, if sufficiently large, produce unstable growth rates in time. This result is new and it is consistent with the known experimental fact that transition to turbulent flow depends not only on Reynolds number but also on the general character of the perturbations which exist in the flow.

The perturbation velocity plots of Section IV represent the first practical look at the function Q. These plots were valuable indicators for adequacy of mesh fineness, that is, N, changes in the nature of the function Q and effects of a nonuniform mesh.

No instabilities were discovered for purely sinusoidal perturbations ($\alpha_R = 0$). This is consistent with the previous investigation of Ref. 11, but should not be assumed for investigations of other angular wave numbers, ($n = 1,2,3, \ldots$).

Adequate numerical accuracy was proven by demonstrating that the solution was virtually independent of the number of mesh points, N, and that it satisfied to a high degree an independent check of the governing differential equation. This procedure should also be carried out in future investigations prior to conducting full scale data runs.

This study suggests that similar, and perhaps even more rewarding results will be obtained for the higher angular wave numbers. Although lengthy, programming is straightforward if approached systematically. The general organization of the programs of Ref. 4 or Ref. 6 should be helpful in this task. It is recommended that the case for n = 1 be undertaken as a follow-on to this study.

The nonuniform computational mesh was shown to be a powerful tool in the reduction of computational time. At the same time, however, the dependence of the mesh offset parameter, λ, on input conditions needs to be investigated further to realize the full potential of this technique.

APPENDIX A

DERIVATION OF VORTICITY TRANSPORT EQUATION COEFFICIENTS

From the change of variable introduced in Ref. 9, the function H for the case n = 0 is expressed by

$$H = rQ \tag{A-1}$$

Taking derivatives

$$DH = rDQ + Q \tag{A-2}$$

$$D^2H = rD^2Q + 2DQ \tag{A-3}$$

$$D^3H = rD^3Q + 3D^2Q \tag{A-4}$$

$$D^4H = rD^4Q + 4D^3Q \tag{A-5}$$

Let the '*' superscript denote element (2,2) of matrices (A1) through (A9) of Ref. 9. Since for n = 0, only the function H was investigated, equations (2-10) become

$$M_4^* D^4H + M_3^* D^3H + M_2^* D^2H + M_1^* DH + M_0^* H$$

$$- \gamma [N_2^* D^2H + N_1^* DH + N_0^* H] = 0 \tag{A-6}$$

Substituting for H, equation (A-6) becomes

$$M_4^*\{rD^4Q+4D^3Q\} + M_3^*\{rD^3Q+3D^2Q\} + M_2^*\{rD^2Q+2DQ\}$$

$$+ M_1^*\{rDQ+Q\} + M_0^*\{rQ\} - \gamma[N_2^*\{rD^2Q+2DQ\}$$

$$+ N_1^*\{rDQ+Q\} + N_0^*\{rQ\}] = 0 \qquad \text{(A-7)}$$

Before proceeding further, it should be noted that the Ref. 9 matrices from which the coefficients for equation (A-7) were taken were obtained from matrices (2-10) through (2-17) of Ref. 6 by means of the following substitutions:

$$U = 2(1 - r^2) \qquad \text{(A-8)}$$

$$t = \alpha^2 \frac{n_2}{r^2} \qquad \text{(A-9)}$$

$$T = \alpha U - \frac{1}{R_e}(\alpha^2 - \frac{n_2}{r^2}) \qquad \text{(A-10)}$$

Defining the new coefficients for equation (A-7) as M_0 through M_4 and N_0 through N_2

$$M_4 = rM_4^* = -\frac{r}{R_e} \qquad \text{(A-11)}$$

$$M_3 = 4M_4^* + rM_3^* = -\frac{6}{R_e} \qquad \text{(A-12)}$$

$$M_2 = 3M_3^* + rM_2^* = r\alpha U - \frac{1}{R_e}\{\frac{3}{r}+2\alpha^2 r\} \qquad \text{(A-13)}$$

$$M_1 = 2M_2^* + rM_1^* = 3\alpha U + \frac{3}{R_e}\{\frac{1}{r^2}-2\alpha^2\} \qquad \text{(A-14)}$$

$$M_0 = M_1^* + rM_0^* = r\alpha^3 U - \frac{\alpha^4 r}{R_e} \qquad \text{(A-15)}$$

$$N_2 = rN_2^* = -r \qquad \text{(A-16)}$$

$$N_1 = 2N_2^* + rN_1^* = -3 \qquad \text{(A-17)}$$

$$N_0 = N_1^* + rN_0^* = -\alpha^2 r \qquad \text{(A-18)}$$

Upon making use of the foregoing substitutions, the governing relation can finally be reduced to the form previously shown in equation (3-1).

APPENDIX B

FINITE DIFFERENCE EQUATIONS

Improved finite difference equations for the boundaries were obtained by not using the virtual point method of Ref. 4 and Ref. 6 and deriving the forms directly from the boundary conditions of Appendix A. The equations thus formed are also of consistent order truncation error, significantly improving the accuracy of the solution [Ref. 8].

Because of a peculiarity in the form of the consistent second order truncation error equations at the axis, a singularity resulted for α equal to zero. Consistent third order truncation error equations eliminated this problem.

From Appendix A, the axis boundary conditions are

$$DQ(0) = 0 \quad \text{and} \quad D^3Q(0) = 0 \tag{B-1}$$

Representing Q by a power series and applying equations (B-1) yields

$$Q(r) = Q(0) + D^2Q(0)\frac{r^2}{2!} + D^4Q(0)\frac{r^4}{4!} + D^5Q(0)\frac{r^5}{5!} + D^6Q(0)\frac{r^6}{6!} + \ldots \tag{B-2}$$

Using five mesh points at $r = \delta, 2\delta, 3\delta, 4\delta$ and 5δ results in the matrix

$$\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{24} & \frac{1}{120} & \frac{1}{720} \\ 1 & 2 & \frac{16}{24} & \frac{32}{120} & \frac{64}{720} \\ 1 & \frac{9}{2} & \frac{81}{24} & \frac{243}{120} & \frac{729}{720} \\ 1 & 8 & \frac{256}{24} & \frac{1024}{120} & \frac{4096}{720} \\ 1 & \frac{25}{2} & \frac{625}{24} & \frac{3125}{120} & \frac{15625}{720} \end{bmatrix} \begin{Bmatrix} Q(0) \\ \delta^2 D^2 Q(0) \\ \delta^4 D^4 Q(0) \\ \delta^5 D^5 Q(0) \\ \delta^6 D^6 Q(0) \end{Bmatrix} + O\delta^7 \qquad \text{(B-3)}$$

Differentiating equation (B-2) and substituting $r = \delta$ gives (in matrix form)

$$\begin{Bmatrix} Q(\delta) \\ \delta DQ(\delta) \\ \delta^2 D^2 Q(\delta) \\ \delta^3 D^3 Q(\delta) \\ \delta^4 D^4 Q(\delta) \end{Bmatrix} = \begin{bmatrix} 1 & \frac{1}{2!} & \frac{1}{4!} & \frac{1}{5!} & \frac{1}{6!} \\ 0 & 1 & \frac{1}{3!} & \frac{1}{4!} & \frac{1}{5!} \\ 0 & 1 & \frac{1}{2!} & \frac{1}{3!} & \frac{1}{4!} \\ 0 & 0 & 1 & \frac{1}{2!} & \frac{1}{3!} \\ 0 & 0 & 1 & 1 & \frac{1}{2!} \end{bmatrix} \begin{Bmatrix} Q(0) \\ \delta^2 D^2 Q(0) \\ \delta^4 D^4 Q(0) \\ \delta^5 D^5 Q(0) \\ \delta^6 D^6 Q(0) \end{Bmatrix} + O\delta^7 \qquad \text{(B-4)}$$

Let [A] and [B] denote the coefficient matrices of equations (B-3) and (B-4) respectively. The values of $Q(0)$, $\delta^2 D^2 Q(0)$, $\delta^4 D^4 Q(0)$, $\delta^5 D^5 Q(0)$ and $\delta^6 D^6 Q(0)$ may be solved for by

$$\begin{Bmatrix} Q(0) \\ \delta^2 D^2 Q(0) \\ \delta^4 D^4 Q(0) \\ \delta^5 D^5 Q(0) \\ \delta^6 D^6 Q(0) \end{Bmatrix} = [A]^{-1} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} + O\delta^7 \qquad \text{(B-5)}$$

Putting equation (B-5) into equation (B-4),

$$\begin{Bmatrix} Q(\delta) \\ \delta DQ(\delta) \\ \delta^2 D^2 Q(\delta) \\ \delta^3 D^3 Q(\delta) \\ \delta^4 D^4 Q(\delta) \end{Bmatrix} = [B][A]^{-1} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} + O\delta^7 \qquad \text{(B-6)}$$

The last line of this set of equations gives

$$D^4 Q(\delta) = \frac{1}{\delta^4}(-.911564626Q_1 + 2.750242955Q_2 - 3.043731779Q_3 + 1.42468416Q_4 - .219630709Q_5) + O\delta^3 \qquad \text{(B-7)}$$

To solve for $D^3 Q(\delta)$, the rightmost column and bottom row are eliminated from matrices [A] and [B] then these new matrices are inserted into equations (B-5) and (B-6).

The bottom line of equation (B-6) will now give the expression for $D^3Q(\delta)$ with a consistent third order truncation error. $D^2Q(\delta)$ and $DQ(\delta)$ were solved for in a similar manner.

$$D^3Q(\delta) = \frac{1}{\delta^3}(1.825165563Q_1 - 3.250331126Q_2 + 1.660927152Q_3 - .235761589Q_4) + O\delta^3 \quad \text{(B-8)}$$

$$D^2Q = \frac{1}{\delta^2}(-\frac{35}{60}Q_1 + \frac{8}{15}Q_2 + \frac{1}{20}Q_3) + O\delta^3 \quad \text{(B-9)}$$

$$DQ = \frac{1}{\delta}(-\frac{2}{3}Q_1 + \frac{2}{3}Q_2) + O\delta^3 \quad \text{(B-10)}$$

Due to the complexity of the boundary conditions, it was decided that consistent third order truncation error equations should also be used at $r = 2\delta$. For this the [B] matrix only need be changed as equation (B-2) is unchanged at this station. The new matrix [B] is formed by differentiating equation (B-2) and making the substitution $r = 2\delta$. Proceeding as for $r = \delta$ gives the following finite difference approximations

$$D^4Q(2\delta) = \frac{1}{\delta^4}(-3.10340136Q_1 + 6.903012634Q_2 - 5.342274053Q_3 + 1.66083577Q_4 - 0.123420797Q_5) + O\delta^3 \quad \text{(B-11)}$$

$$D^3Q(2\delta) = \frac{1}{\delta^3}(.868874172Q_1 - .937748345Q_2 - .254304636Q_3 + .323178808Q_4) + O\delta^3 \quad \text{(B-12)}$$

$$D^2Q(2\delta) = \frac{1}{\delta^2}\left(\frac{11}{12}Q_1 - \frac{28}{15}Q_2 + \frac{19}{20}Q_3\right) + O\delta^3 \qquad \text{(B-13)}$$

$$DQ(2\delta) = \frac{1}{\delta}\left(-\frac{4}{3}Q_1 + \frac{4}{3}Q_2\right) + O\delta^3 \qquad \text{(B-14)}$$

It should also be noted that the value of Q at r = 0 may be solved for from the top line of equations (B-5)

$$Q(0) = (1.795918367Q_1 - 1.24781341Q_2 + .606413994Q_3 - .177842566Q_4 + .023323615Q_5) + O\delta^3 \qquad \text{(B-15)}$$

The central difference equations given by Ref. 6 were already consistent second order truncation error equations as confirmed by Ref. 8 and were retained.

For the wall, the clamped end, consistent second order equations (5) through (8) of Table II, Ref. 8 were modified for the "right boundary" using the procedure given in Section 5 of that reference.

$$D^4Q(1-\delta) = \frac{1}{\delta^4}\left(-\frac{1}{4}Q_{N-3} + \frac{8}{3}Q_{N-2} - 9Q_{N-1} + 16Q_N\right) + O\delta^2 \qquad \text{(B-16)}$$

$$D^3Q(1-\delta) = \frac{1}{\delta^3}\left(-\frac{1}{3}Q_{N-2} + 3Q_N\right) + O\delta^2 \qquad \text{(B-17)}$$

$$D^2Q(1-\delta) = \frac{1}{\delta^2}\left(Q_{N-1} - 2Q_N\right) + O\delta^2 \qquad \text{(B-18)}$$

$$DQ(1-\delta) = \frac{1}{\delta}\left(-\frac{1}{2}Q_{N-1}\right) + O\delta^2 \qquad \text{(B-19)}$$

Since the wall finite difference approximations were of only second order truncation error, the approximations for DQ through D^4Q at $r = 1-2\delta$ were obtained directly from the central difference equations with $Q(1) = 0$.

$$D^4Q(1-2\delta) = \frac{1}{\delta^4}(Q_{N-3} - 4Q_{N-2} + 6Q_{N-1} - 4Q_N) + O\delta^2 \quad \text{(B-20)}$$

$$D^3Q(1-2\delta) = \frac{1}{\delta^3}(-\frac{1}{2}Q_{N-3} + Q_{N-2} - Q_N) + O\delta^2 \quad \text{(B-21)}$$

$$D^2Q(1-2\delta) = \frac{1}{\delta^2}(Q_{N-2} - 2Q_{N-1} + Q_N) + O\delta^2 \quad \text{(B-22)}$$

$$DQ(1-2\delta) = \frac{1}{\delta}(-\frac{1}{2}Q_{N-2} + \frac{1}{2}Q_N) + O\delta^2 \quad \text{(B-23)}$$

APPENDIX C

NONUNIFORM MESH

To control the distribution of a fixed number of mesh points, a change of the independent variable from r to η was performed.

$$Q = Q(\eta) \tag{C-1}$$

$$r = r(\eta) \tag{C-2}$$

The derivative with respect to r becomes

$$D = (D^* r)^{-1} D^* \tag{C-3}$$

where

$$D^* = \frac{d}{d\eta} \quad \text{and} \quad D = \frac{d}{dr} \tag{C-4}$$

DQ, D^2Q ... can now be expressed in terms of the new independent variable, η.

$$DQ = (D^* r)^{-1} D^* Q \tag{C-5}$$

$$D^2Q = D(DQ) = (D^* r)^{-1} D^* (DQ)$$

$$= (D^* r)^{-2} D^{*2} Q - (D^* r)^{-3} (D^{*2} r) D^* Q \tag{C-6}$$

$$D^3Q = D(D^2Q) = (D^*R)^{-1}D^*(D^2Q)$$

$$= (D^*r)^{-3}D^{*3}Q - 3(D^*r)^{-4}(D^{*2}r)D^{*2}Q$$

$$- [(D^*r)^{-4}(D^{*3}r) - 3(D^*r)^{-5}(D^{*2}r)^2]DQ \qquad (C\text{-}7)$$

$$D^4Q = D(D^3Q) = (D^*r)^{-1}D^*(D^3Q)$$

$$= (D^*r)^{-4}D^{*4}Q - 6(D^*r)^{-5}(D^{*2}r)D^{*3}Q$$

$$+ [15(D^*r)^{-6}(D^{*2}r) - 4(D^*r)^{-5}(D^{*3}r)]D^{*2}Q$$

$$- [15(D^*r)^{-7}(D^{*2}r)^3 - 10(D^*r)^{-6}(D^{*2}r)(D^{*3}r)$$

$$+ (D^*r)^{-5}(D^{*4}r)]DQ \qquad (C\text{-}8)$$

The derivatives of Q with respect to r can now be written

$$DQ = f_{11}D^*Q \qquad (C\text{-}9)$$

$$D^2Q = f_{22}D^*2Q + f_{21}D^*Q \qquad (C\text{-}10)$$

$$D^3Q = f_{33}D^{*3}Q + f_{32}D^{*2}Q + f_{31}D^*Q \qquad (C\text{-}11)$$

$$D^4Q = f_{44}D^{*4}Q + f_{43}D^{*3}Q + f_{42}D^{*2}Q + f_{41}D^*Q \qquad (C\text{-}12)$$

where

$$f_{11} = (D^*r)^{-1} \qquad (C\text{-}13)$$

$$f_{22} = (D^{*}r)^{-2} \tag{C-14}$$

$$f_{21} = -(D^{*}r)^{-3}(D^{*2}r) \tag{C-15}$$

$$f_{33} = (D^{*}r)^{-3} \tag{C-16}$$

$$f_{32} = -3(D^{*}r)^{-4}(D^{*2}r) \tag{C-17}$$

$$f_{31} = 3(D^{*}r)^{-5}(D^{*2}r)^{2} - (D^{*}r)^{-4}(D^{*3}r) \tag{C-18}$$

$$f_{44} = (D^{*}r)^{-4} \tag{C-19}$$

$$f_{43} = -6(D^{*}r)^{-5}(D^{*2}r) \tag{C-20}$$

$$f_{42} = 15(D^{*}r)^{-6}(D^{*2}r)^{2} - 4(D^{*}r)^{-5}(D^{*3}r) \tag{C-21}$$

$$f_{41} = -15(D^{*}r)^{-7}(D^{*2}r)^{3} + 10(D^{*}r)^{-6}(D^{*2}r)(D^{*3}r) - (D^{*}r)^{-5}(D^{*4}r) \tag{C-22}$$

Substituting equations (C-9) through (C-12) into the vorticity transport equation (A-6) yields

$$M_4^{*}D^{*4}Q + M_3^{*}D^{*3}Q + M_2^{*}D^{*2}Q + M_1^{*}D^{*}Q + M_0^{*}Q - \gamma[N_2^{*}D^{*2}Q + N_1^{*}D^{*}Q + N_0^{*}Q] = 0 \tag{C-23}$$

where

$$M_4^* = M_4 f_{44} \tag{C-24}$$

$$M_3^* = M_4 f_{43} + M_3 f_{33} \tag{C-25}$$

$$M_2^* = M_4 f_{42} + M_3 f_{32} + M_2 f_{22} \tag{C-26}$$

$$M_1^* = M_4 f_{41} + M_3 f_{31} + M_2 f_{21} \tag{C-27}$$

$$M_0^* = M_0 \tag{C-28}$$

$$N_2^* = N_2 f_{22} \tag{C-29}$$

$$N_1^* = N_2 f_{21} + N_1 f_{11} \tag{C-30}$$

$$N_0^* = N \tag{C-31}$$

In order to concentrate the mesh points at the axis, the function

$$r = 1 - C \tanh \lambda(1-\eta) \tag{C-32}$$

was chosen where λ is a parameter controlling the degree of concentration of mesh points near the axis. Equation (C-32) must satisfy the two conditions

$$r = 0 \quad \text{at} \quad \eta = 0 \tag{C-33}$$

and

$$r = 1 \quad \text{at} \quad \eta = 1 .$$

Substituting equation (C-33) into (C-32) gives

$$C = 1/\tanh \lambda . \tag{C-35}$$

Computing derivatives

$$D^{*}r = C\lambda/\cosh^{2}\lambda(1-\eta) \tag{C-36}$$

$$D^{*2}r = 2C\lambda^{2}[\tanh\lambda(1-\eta)/\cosh^{2}\lambda(1-\eta)] \tag{C-37}$$

$$D^{*3}r = -2C\lambda^{3}\{[1-2\sinh^{2}\lambda(1-\eta)]/\cosh^{4}\lambda(1-\eta)\} \tag{C-38}$$

$$D^{*4}r = 8C\lambda^{4}[\tanh^{3}\lambda(1-\eta)/\cosh^{2}\lambda(1-\eta)] \tag{C-39}$$

To shift the mesh point concentration to the wall, the function

$$r = C \tanh \lambda\eta \tag{C-40}$$

was selected. Satisfying equations (C-33) and (C-34) for this equation also gives equation (C-35). The derivatives

of (C-40) are given by equations (C-36) through (C-39) if η is substituted for all occurrences of $(1-\eta)$ and the signs of equations (C-37) and (C-39) are reversed. Figures C-1 and C-2 show equations (C-32) and (C-40) for four selected values of the parameter λ.

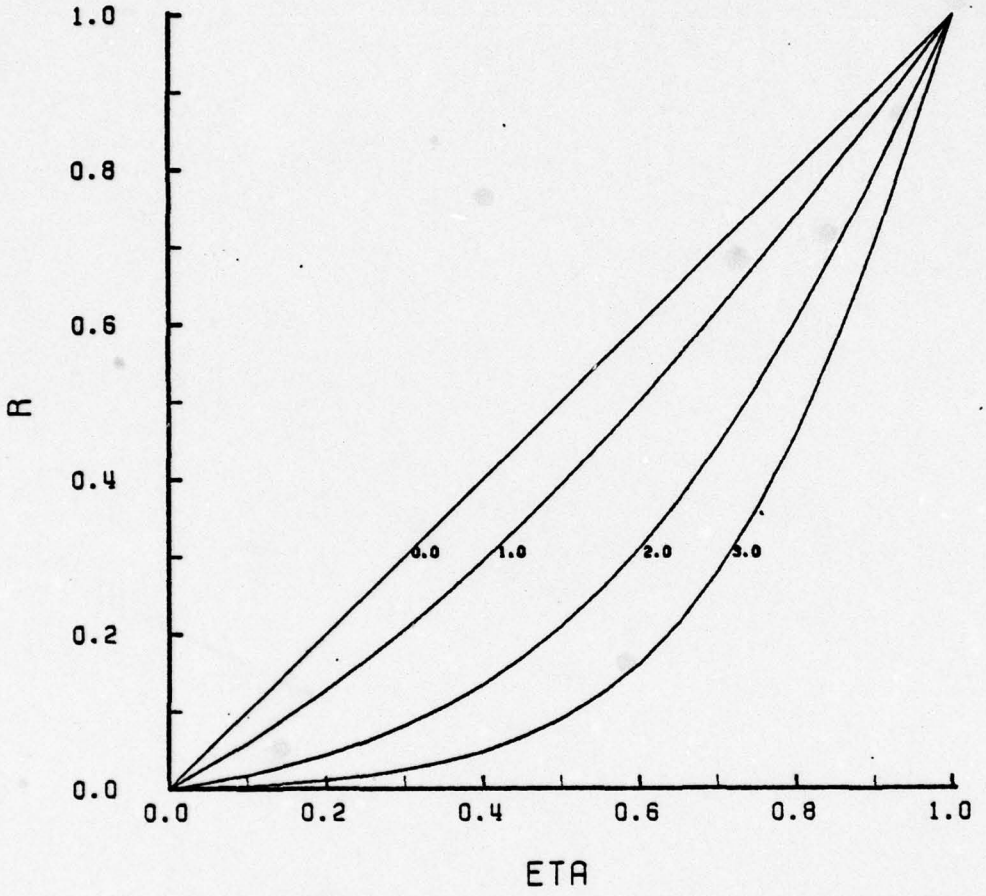


FIGURE C-1. R Versus η for Four Selected Values of Lambda - Axis Offset

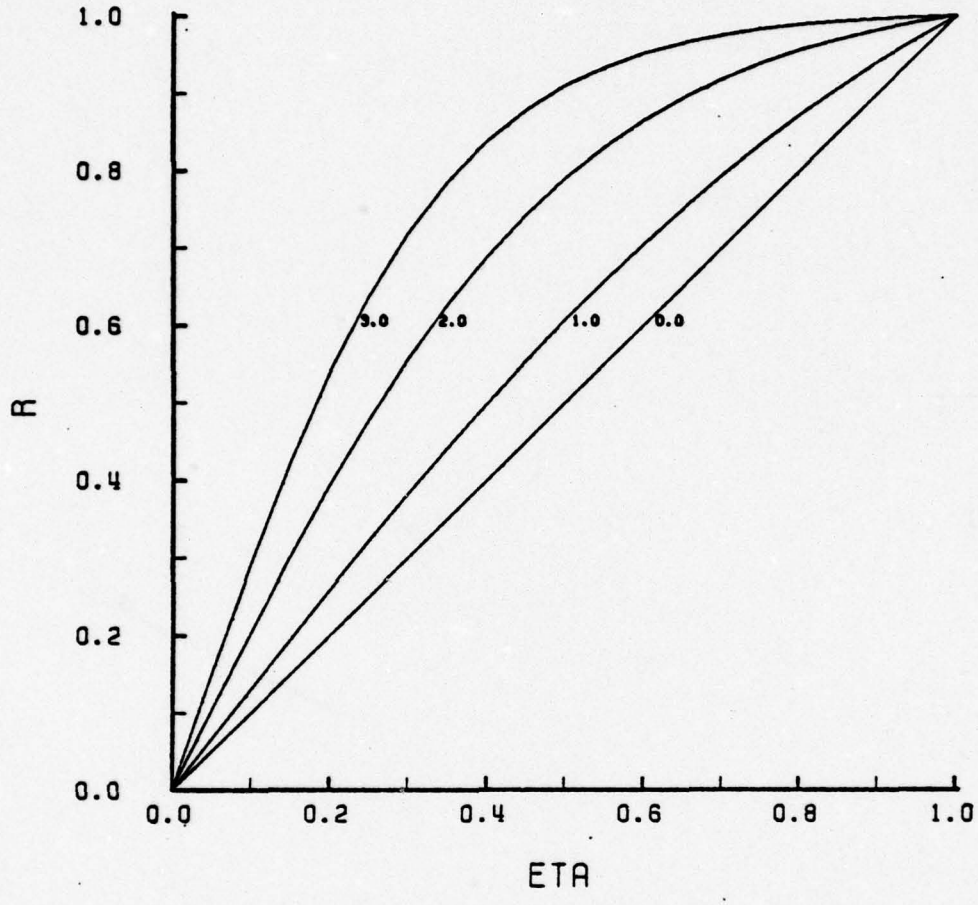


FIGURE C-2. R Versus η for Four Selected Values of Lambda - Wall Offset

APPENDIX D

DERIVATION OF PERTURBATION VELOCITIES

From Ref. 4, Appendix E, equations E-6 through E-8:

$$\begin{Bmatrix} u(r) \\ v(r) \\ w(r) \end{Bmatrix} = [A]\bar{W} + [B]D\bar{W} \tag{D-1}$$

$$= \begin{bmatrix} 0 & -\frac{\beta}{r} & \frac{1}{r} \\ \frac{\beta}{r} & 0 & -\alpha \\ 0 & \alpha & 0 \end{bmatrix} \begin{Bmatrix} F \\ G \\ H \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} DF \\ DG \\ DG \end{Bmatrix} \tag{D-2}$$

For this case $\beta = ni = 0$ and $F = DF = 0$. Restricting the investigation to the function H for the reason expressed in Section I and solving for u(r) gives

$$u(r) = \frac{H}{r} + DH \tag{D-3}$$

Performing the change of variable

$$H = rQ \tag{D-4}$$

$$DH = Q + rDQ \tag{D-5}$$

$$u(r) = \frac{rQ}{r} + (Q + rDQ) = 2Q + rDQ \qquad \text{(D-6)}$$

In order to implement this derivation in a numerical analysis, equation (D-6) was rewritten as

$$u_i = 2Q_i + r_i DQ_i \qquad \text{(D-7)}$$

Performing the change of independent variable (Appendix C) to accommodate a nonuniform mesh

$$Q_i = Q(\eta_i) \qquad \text{(D-8)}$$

$$r_i = r(\eta_i) \qquad \text{(D-9)}$$

$$DQ_i = (D^* r_i)^{-1} D^* Q(\eta_i) \qquad \text{(D-10)}$$

Substituting equations (D-8), (D-9) and (D-10) into equation (D-7) gives

$$u_i = 2Q(\eta_i) + r(\eta_i) D^* r_i)^{-1} D^* Q(\eta_i) \qquad \text{(D-11)}$$

For the axis offset nonuniform mesh, $r(\eta)$ is given by equation (C-32) and $(D^* r)$ by equation (C-36). Substituting into equation (D-11) using equation (C-35) results in

$$u_i = 2Q(\eta_i) + \{1 - \frac{\tanh[\lambda(1-\eta_i)]}{\tanh\lambda}\}\{\frac{\cosh^2[\lambda(1-\eta_i)]}{C\lambda}\}D^*Q(\eta_i)$$

$$= 2Q(\eta_i)+\{1 - \frac{\tanh[\lambda(1-\eta_i)]}{\tanh\lambda} \quad \frac{\tanh\lambda\cosh^2[\lambda(1-\eta_i)]}{\lambda}\}D^*Q(\eta_i)$$

(D-12)

For the wall offset mesh, equation (C-40) is substituted for equation (C-32) and all occurrences of the term $1-\eta_i$ are replaced by the term η_i.

The value of u at the axis (u_o) and at the wall (u_{N+1}) were solved for by using the boundary conditions specified in Ref. 9, namely

$$Q(1) = 0 \qquad \text{(D-13)}$$

$$DQ(1) = 0 \qquad \text{(D-14)}$$

$$DQ(0) = 0 \qquad \text{(D-15)}$$

$$D^3Q(0) = 0 \qquad \text{(D-16)}$$

From equations (D-13) and (D-14), using equation (D-7) it is obvious that

$$u_{N+1} = 0 \qquad \text{(D-17)}$$

and from equations (D-15) and (D-7), it is similarly found that

$$u_o = 2Q(0) , \qquad \text{(D-18)}$$

where the finite difference approximation for Q(0) is given by equation (B-15).

```
C.....PIPO  10
C                                                                      PIPO  20
C     PROGRAM PIPEO  (CP/CMS VERSION)                                  PIPO  30
C                                                                      PIPO  40
C        PROGRAM TO INVESTIGATE FLOW STABILITY AND CHARACTERISTICS     PIPO  50
C              FOR THE 3-D CYLINDRICAL FLOW PROBLEM                    PIPO  60
C                   NI=0;  FOR THE FUNCTION Q                          PIPO  70
C                                                                      PIPO  80
C        TO OBTAIN A FLOW CHART OF THIS PROGRAM, CONSULT NAVAL         PIPO  90
C        POSTGRADUATE SCHOOL TECHNICAL NOTE TN 0141-25, "USER'S        PIPO 100
C        GUIDE TO THE PROGRAMMING AIDS LIBRARY",UNDER PROGRAM          PIPO 110
C        FLOWCH.                                                       PIPO 120
C                                                                      PIPO 130
C.....PIPO 140
C                                                                      PIPO 150
C-----PIPC 160
      IMPLICIT REAL*8(A-H,O-Z)                                         PIPO 170
      DIMENSION AIPLT(20), REYPLT(20)                                  PIPO 180
      COMPLEX *16A,G                                                   PIPO 190
      INTEGER *4CLOCK(6)                                               PIPO 200
      COMMON /COEFNT/ A,G,REY,DELR,AMDA                                PIPO 210
C-----PIPO 220
C                                                                      PIPO 230
C                                                                      PIPO 240
C     INPUT DESIRED MODE NUMBER                                        PIPO 250
C                                                                      PIPO 260
C                                                                      PIPO 270
      WRITE (6,6)                                                      PIPO 280
      READ (5,7) MODENO                                                PIPO 290
      CALL IXCLOK (CLOCK)                                              PIPO 300
      WRITE (6,8) CLOCK(3),CLOCK(4)                                    PIPO 310
      IF (MODENO.EQ.2) GO TO 2                                         PIPO 320
C                                                                      PIPO 330
C                                                                      PIPO 340
C**********PIPO 350
C                                                                      PIPO 360
C     CALCULATE STABILITY AT A POINT (MODENO = 1 & 3)                  PIPO 370
C                                                                      PIPO 380
C        OUTPUTS DATA TO FILE FT02F001 COMPATIBLE WITH PROGRAM EIGFCN & PIPO 390
C        PRINTS THE VALUE OF THE LEAST STABLE EIGENVALUE(GAMMA*) AT THE PIPO 400
C        CONSOLE FOR EACH SET OF INPUT CONDITIONS.  DATA IS ENTERED IN PIPO 410
C        'D' FORMAT WITH A ZERO EXPONENT (I.E.,1.D0,1150.D0,-.5D0,ETC.).PIPO 420
C        RUN IS TERMINATED WHEN A NEGATIVE VALUE FOR REYNOLDS NUMBER IS PIPO 430
C        ENTERED.  THE TYPE OF MESH OFFSET IS DETERMINED BY THE VALUE   PIPO 440
C        OF LAMBDA.  IF LAMBDA < -1.D-10, KSET IS SET EQUAL TO -1; IF   PIPO 450
C        LAMDA > 1.D-10, KSET IS SET EQUAL TO 1; CTHERWISE KSET = 0.    PIPO 460
C        IF MODENO = 1, THE EIGENVECTOR CORRESPONDING TO THE LEAST      PIPO 470
C        STABLE EIGENVALUE (GAMMA*) WILL BE WRITTEN TO FILE FTC2F001.   PIPO 480
```

```
C        IF MODENO = 3, THIS OUTPUT IS INHIBITED.  MODENO MUST BE        PIPO 490
C        SET EQUAL TO ONE TO GENERATE CORRECT DATA FOR PROGRAM EIGFCN.   PIPO 500
C                                                                        PIPO 510
C*****************************************************************     PIPO 520
C                                                                        PIPO 530
    1 WRITE (6,9)                                                        PIPO 540
      READ (5,11) AMDA                                                   PIPO 550
      KSET = 0                                                           PIPO 560
      IF (AMDA.LT.-1.D-10) KSET=-1                                       PIPO 570
      IF (AMDA.GT.1.D-10) KSET=1                                         PIPO 580
      WRITE (6,10)                                                       PIPO 590
      READ (5,11) AR                                                     PIPO 600
      WRITE (6,12)                                                       PIPO 610
      READ (5,11) AI                                                     PIPO 620
      WRITE (6,13)                                                       PIPO 630
      READ (5,11) REY                                                    PIPO 640
      IF (REY.LE.0.D0) GO TO 5                                           PIPO 650
      CALL STAB (AR,AI,GRMAX,KSET,MODENO)                                PIPO 660
      WRITE (6,14) GRMAX                                                 PIPO 670
      GO TO 1                                                            PIPO 680
C                                                                        PIPO 690
C                                                                        PIPO 700
C*****************************************************************     PIPO 710
C                                                                        PIPO 720
C     COMPUTE STABILITY MAP (MODENO = 2)                                 PIPO 730
C                                                                        PIPO 740
C        COMPUTES A STABILITY MAP AT MESH POINTS ESTABLISHED BY          PIPO 750
C        THE FOLLOWING PARAMTERS READ FROM FILE FT01F001:                PIPO 760
C                                                                        PIPO 770
C             NXSTP - NO OF MESH PTS IN X-DIRECTION                      PIPO 780
C             NYSTP - NO OF MESH PTS IN Y-DIRECTION                      PIPO 790
C             N     - DIMENSION OF MATRICES X & Y IN SUBROUTINE STAB     PIPO 800
C             DELAR - MAGNITUDE OF THE X-DIRECTION STEP                  PIPO 810
C             DELAI - MAGNITUDE OF THE Y-DIRECTION STEP                  PIPO 820
C             REY   - REYNOLDS NUMBER                                    PIPO 830
C                                                                        PIPO 840
C        **NOTE - RUN TIME IS LONG IN THIS MODE, SO CP/CMS               PIPO 850
C        RUNS SHOULD BE LIMITED TO 10X10 MESHES OR LESS WITH             PIPO 860
C        N = 31.  LARGER RUNS SHOULD BE MADE UNDER BATCH.                PIPO 870
C        LITTLE OS MAY ALSO BE USED AS LESS THAN 18CK OF                 PIPO 880
C        CORE IS REQUIRED IN THIS MODE FOR N <= 47.                      PIPO 890
C                                                                        PIPO 900
C        **NOTE - TO RUN THE MAPPING PORTION UNDER OS OR LITTLE OS,      PIPO 910
C        PERFORM THE FOLLOWING:                                          PIPO 920
C                                                                        PIPO 930
C             1)RETAIN ALL MAIN PROGRAM SECTIONS BRACKETED BY '----'.    PIPO 940
C             2)CHANGE ALL READ DEVICES TO '5' VICE '1' IN MAIN PROGRAM. PIPO 950
C             3)CHANGE THE DEVICE CODE OF THE LAST WRITE STATEMENT       PIPO 960
```

```
C              PRIOR TO THE 'STOP' IN THE MAIN PROGRAM TO '7' VICE '3'.  PIPO 970
C              CHANGE ALL OTHER MAIN PROGRAM WRITE STATEMENTS TO DEVICE  PIPO 980
C              CODE '6'.                                                 PIPO 990
C            4)IMMEDIATELY AFTER STATEMENT 116 IN MAIN PROGRAM, INSERT   PIP01000
C              THE FOLLOWING: 'MODENO = 2'                               PIP01010
C            5)DELETE PORTION BETWEEN '---' MARKINGS IN                  PIP01020
C              SUBROUTINE STAB.                                          PIP01030
C                                                                        PIP01040
C       APPROXIMATE RUNNING TIMES ARE 490 MINUTES(CPU) FOR N=47 AND A    PIP01050
C       21 X 21 MESH AND 430 MINUTES(CPU) FOR N=31 AND A 41 X 41 MESH.   PIP01060
C                                                                        PIP01070
C       NXSTP MUST EQUAL NYSTP IF PLOT DESIRED BY PROGRAM STBCONT.       PIP01080
C                                                                        PIP01090
C*************************************************************************PIP01100
C                                                                        PIP01110
C-------------------------------------------------------------------------PIP01120
    2 READ (1,15) NXSTP,NYSTP,N,DELAR,DELAI,REY                          PIP01130
      READ (1,16) ARSTRT,AISTRT                                          PIP01140
      READ (1,16) AMDA                                                   PIP01150
      KSET = 0                                                           PIP01160
      IF (AMDA.LT.-1.D-10) KSET=-1                                       PIP01170
      IF (AMDA.GT.1.D-10) KSET=1                                         PIP01180
      AR = ARSTRT                                                        PIP01190
      WRITE (6,17) REY                                                   PIP01200
      WRITE (3,18) NXSTP                                                 PIP01210
C                                                                        PIP01220
      DO 4 I=1,NXSTP                                                     PIP01230
      AI = AISTRT                                                        PIP01240
C                                                                        PIP01250
      DO 3 J=1,NYSTP                                                     PIP01260
      CALL STAB (AR,AI,GRMAX,KSET,MODENO)                                PIP01270
      WRITE (6,19) AR,AI,GRMAX                                           PIP01280
      WRITE (3,20) AR,AI,GRMAX                                           PIP01290
    3 AI = AI+DELAI                                                      PIP01300
C                                                                        PIP01310
    4 AR = AR+DELAR                                                      PIP01320
C                                                                        PIP01330
C                                                                        PIP01340
C-------------------------------------------------------------------------PIP01350
    5 CALL IXCLOK (CLOCK)                                                PIP01360
      WRITE (6,21) CLOCK(3),CLOCK(4)                                     PIP01370
C-------------------------------------------------------------------------PIP01380
      STOP                                                               PIP01390
C                                                                        PIP01400
    6 FORMAT ('0',5X,'INPUT MODENO')                                     PIP01410
    7 FORMAT (I1)                                                        PIP01420
    8 FORMAT ('0',5X,'START TIME =',2A4)                                 PIP01430
    9 FORMAT ('0',5X,'INPUT LAMBDA')                                     PIP01440
```

```
   10 FORMAT (' ',5X,'INPUT ALPHA REAL')                                  PIP01450
   11 FORMAT (D20.10)                                                     PIP01460
   12 FORMAT (' ',5X,'INPUT ALPHA IMAG')                                  PIP01470
   13 FORMAT (' ',5X,'INPUT REYNOLDS NO')                                 PIP01480
   14 FORMAT (' ',5X,'STAB=',D20.10)                                      PIP01490
C-----------------------------------------------------------------------PIP01500
   15 FORMAT (3I2,3D20.10)                                                PIP01510
   16 FORMAT (2D20.10)                                                    PIP01520
   17 FORMAT (' ','REYNOLDS NUMBER = ',F8.1,/,10X,'AR',18X,               PIP01530
     1   'AI',16X,'STAB',//)                                             PIP01540
   18 FORMAT (I2)                                                         PIP01550
   19 FORMAT (' ',3D20.10)                                                PIP01560
   20 FORMAT (3E20.10)                                                    PIP01570
C-----------------------------------------------------------------------PIP01580
   21 FORMAT ('0','STOP TIME=',2A4)                                       PIP01590
C-----------------------------------------------------------------------PIP01600
      END                                                                 PIP01610
C-----------------------------------------------------------------------PIP01620
C
C
C.....SUBROUTINE STAB(AR,AI,GRMAX,KSET,MODENO).....STAB  10
C                                                                         STAB  20
C     PURPOSE                                                             STAB  30
C                                                                         STAB  40
C        RETURNS THE REAL PORTION OF THE LEAST STABLE EIGENVALUE          STAB  50
C        FOR THE GIVEN INPUT CONDITIONS. THIS VALUE DETERMINES THE        STAB  60
C        STABILITY OF THE FLOW.                                           STAB  70
C                                                                         STAB  80
C     USAGE                                                               STAB  90
C                                                                         STAB 100
C        CALL STAB(AR,AI,GRMAX,KSET,MODENO)                               STAB 110
C                                                                         STAB 120
C     DESCRIPTION OF PARAMETERS                                           STAB 130
C                                                                         STAB 140
C        AR - THE REAL PART OF ALPHA                                      STAB 150
C                                                                         STAB 160
C        AI - THE IMAGINARY PART OF ALPHA.                                STAB 170
C                                                                         STAB 180
C        GRMAX - THE REAL PORTION OF THE LEAST STABLE EIGENVALUE.         STAB 190
C                THIS VALUE IS RETURNED TO THE CALLING PROGRAM.           STAB 200
C                                                                         STAB 210
C        N - THE NUMBER OF INTERIOR MESH POINTS                           STAB 220
C                                                                         STAB 230
C        KSET - AN INTEGER DENOTING THE TYPE OF MESH OFFSET               STAB 240
C               USED.                                                     STAB 250
C                 KSET = -1 FOR WALL OFFSET                               STAB 260
C                 KSET =  0 FOR UNIFORM MESH                              STAB 270
C                 KSET =  1 FOR AXIS OFFSET                               STAB 280
```

```
C                                                                       STAB  290
CC         MODENO - AN INTEGER CONTROLLING THE OUTPUT OF                STAB  300
CC                  OF EIGENVECTORS TO FILE FT02F001.  IF MODENO         STAB  310
CC                  IS EQUAL TO +1, EIGENVECTORS ARE OUTPUT;            STAB  320
CC                  OTHERWISE OUTPUT IS INHIBITED.                      STAB  330
CC                                                                      STAB  340
CC     OTHER ROUTINES NEEDED                                            STAB  350
CC                                                                      STAB  360
CC        MSET2,CDMTIN,MULM,DSPLIT,EHESSC,ELRH2C,EBALAC,EBBCKC          STAB  370
CC                                                                      STAB  380
CC*****************************************************************   STAB  390
C                                                                       STAB  400
      SUBROUTINE STAB (AR,AI,GRMAX,KSET,MODENO)                         STAB  410
      IMPLICIT REAL*8(A-H,O-Z)                                          STAB  420
      COMPLEX *16A,G                                                    STAB  430
      COMPLEX *16CQM1E1,CQM2E1                                          STAB  440
C                                                                       STAB  450
CC        NOTE--CHANGE DIMENSIONS FROM HERE THROUGH 'N = ' FOR          STAB  460
CC        NEW NMESH. CP/CMS MAX NMESH IS 79. LOGIN WITH 520K OF CORE.   STAB  470
C                                                                       STAB  480
      REAL *8GR(79),GI(79),ZR(79,79),ZI(79,79),RADIUS(79),CVEC(79)      STAB  490
      COMPLEX *16XMAT(79,79),YMAT(79,79),WV(79)                         STAB  500
      DIMENSION IVEC(79)                                                STAB  510
      COMMON /COEFNT/ A,G,REY,DELR,AMDA                                 STAB  520
      EXTERNAL CQM1E1,CQM2E1                                            STAB  530
C                                                                       STAB  540
C                                                                       STAB  550
      A = DCMPLX(AR,AI)                                                 STAB  560
      MDIM = 79                                                         STAB  570
      N = 79                                                            STAB  580
C                                                                       STAB  590
CC                                                                      STAB  600
CC        SET UP THE CENTRAL DIFFERENCE APPROXIMATION AT EACH POINT IN  STAB  610
CC        THE MESH FOR THOSE TERMS IN THE VORTICITY TRANSPORT EQUATION  STAB  620
CC        WHICH DO NOT CONTAIN GAMMA AS A FACTOR.                       STAB  630
CC                                                                      STAB  640
C                                                                       STAB  650
      CALL MSET2 (XMAT,N,MDIM,CQM1E1,KSET)                              STAB  660
C                                                                       STAB  670
CC                                                                      STAB  680
CC        SET UP THE CENTRAL DIFFERENCE APPROXIMATION AT EACH POINT IN  STAB  690
CC        THE MESH FOR THOSE TERMS IN THE VORTICITY TRANSPORT EQUATION  STAB  700
CC        WHICH CONTAIN GAMMA AS A FACTOR.                              STAB  710
CC                                                                      STAB  720
C                                                                       STAB  730
      CALL MSET2 (YMAT,N,MDIM,CQM2E1,KSET)                              STAB  740
C                                                                       STAB  750
C                                                                       STAB  760
```

```
C          INVERT THE RESULTING ARRAY                                          STAB 770
CC                                                                             STAB 780
C                                                                              STAB 790
      CALL CDMTIN (MDIM,YMAT,MDIM,IERR)                                        STAB 800
C                                                                              STAB 810
CC                                                                             STAB 820
CC         PREMULTIPLY XMAT BY THE INVERSE OF YMAT TO CONVERT THE PROBLEM      STAB 830
CC         TO THE STANDARD FORM:                                               STAB 840
CC                              (A - GAMMA*I)*X = 0                            STAB 850
CC                                                                             STAB 860
C                                                                              STAB 870
      CALL MULM (YMAT,XMAT,MDIM,MDIM,WV)                                       STAB 880
C                                                                              STAB 890
CC                                                                             STAB 900
CC         SPLIT THE RESULTING ARRAY INTO REAL (XMAT) AND IMAGINARY (YMAT)     STAB 910
CC         PARTS.                                                              STAB 920
CC                                                                             STAB 930
C                                                                              STAB 940
      CALL DSPLIT (MDIM,MDIM,YMAT,XMAT,YMAT)                                   STAB 950
C                                                                              STAB 960
CC                                                                             STAB 970
CC         CALCULATE THE EIGENVALUES (GAMMA) FOR THE EQUATION.                 STAB 980
CC                                                                             STAB 990
C                                                                              STAB1000
      CALL EBALAC (XMAT,YMAT,MDIM,MDIM,KBND,LBND,DVEC)                         STAB1010
      CALL EHESSC (XMAT,YMAT,KBND,LBND,MDIM,MDIM,IVEC)                         STAB1020
      CALL ELRH2C (XMAT,YMAT,KBND,LBND,MDIM,MDIM,GR,GI,ZR,ZI,IVEC,INERR,       STAB1030
     1IER)                                                                     STAB1040
      CALL EBBCKC (ZR,ZI,MDIM,MDIM,KBND,LBND,MDIM,DVEC)                        STAB1050
      IF (INERR.NE.0) WRITE (6,4) INERR,IER                                    STAB1060
C                                                                              STAB1070
CC    DETERMINE LARGEST GAMMA REAL                                             STAB1080
C                                                                              STAB1090
      NEIG = 1                                                                 STAB1100
      GRMAX = -1.0D02                                                          STAB1110
C                                                                              STAB1120
      DO 1 I=1,MDIM                                                            STAB1130
      IF (GR(I).GT.GRMAX) NEIG=I                                               STAB1140
      IF (GR(I).GT.GRMAX) GRMAX=GR(I)                                          STAB1150
    1 CONTINUE                                                                 STAB1160
C                                                                              STAB1170
      GRMAX = GRMAX+AR                                                         STAB1180
C-------------------------------------------------------------------------     STAB1190
      DEL10 = 1.D0/DFLOAT(N+1)                                                 STAB1200
      WRITE (2,5) MDIM,REY,AR,AI                                               STAB1210
      WRITE (2,7) AMDA,GR(NEIG),GI(NEIG)                                       STAB1220
      WRITE (2,6) KSET                                                         STAB1230
      IF (MODENO.NE.1) GO TO 3                                                 STAB1240
```

```
C                                                                       STAB1250
      DO 2 I=1,MDIM                                                     STAB1260
      RADIUS(I) = DFLOAT(I)*DEL10                                       STAB1270
      WRITE (2,7) RADIUS(I),ZR(I,NEIG),ZI(I,NEIG)                       STAB1280
C-----------------------------------------------------------------------STAB1290
    2 CONTINUE                                                          STAB1300
C                                                                       STAB1310
    3 RETURN                                                            STAB1320
C                                                                       STAB1330
    4 FORMAT ('0* * * ERROR NUMBER',I7,' ON EIGENVALUE',               STAB1340
     1      I7,' * * *',///)                                           STAB1350
    5 FORMAT (I2,3D20.10)                                               STAB1360
    6 FORMAT (I2)                                                       STAB1370
    7 FORMAT (F15.7,2(1PD20.10))                                        STAB1380
      END                                                               STAB1390
C
CC.....................SUBROUTINE MSET1(ETA,CQM1,CQM2,KSET).............MST1   10
CC                                                                      MST1   20
CC     PURPOSE                                                          MST1   30
CC                                                                      MST1   40
CC        MSET1 GENERATES THE COEFFICIENTS FOR THE FINITE DIFFERENCE    MST1   50
CC                                                                      MST1   70
CC        APPROXIMATION OF THE COMPONENT Q.                             MST1   60
CC     USAGE                                                            MST1   80
CC                                                                      MST1   90
CC        CALL MSET1(ETA,CQM1,CQM2,KSET)                                MST1  100
CC                                                                      MST1  110
CC     DESCRIPTION OF PARAMETERS                                        MST1  120
CC                                                                      MST1  130
CC        ETA -  INDEPENDENT VARIABLE REPLACING R                       MST1  140
CC               IN NONUNIFORM MESH.                                    MST1  150
CC                                                                      MST1  160
CC        CQM1 - COEFFICIENTS OF Q AND ITS DERIVATIVES                  MST1  170
CC               IN THE FINITE DIFFERENCE APPROXIMATION OF              MST1  180
CC               THE NON-GAMMA TERMS. CQM1(1) IS THE                    MST1  190
CC               COEFFICIENT FOR Q, CQM1(2) IS THE COEF-                MST1  200
CC               FICIENT FOR DQ, AND SO ON TO CQM1(5).                  MST1  210
CC                                                                      MST1  220
CC        CQM2 - SAME AS CQM1 EXCEPT GAMMA TERMS AND                    MST1  230
CC               DIMENSIONED 3 INSTEAD OF 5.                            MST1  240
CC                                                                      MST1  250
CC        KSET - MESH OFFSET PARAMETER AS DESCRIBED FOR                 MST1  260
CC               SUBROUTINE STAB.                                       MST1  270
CC                                                                      MST1  280
CC     OTHER ROUTINES NEEDED                                            MST1  290
CC                                                                      MST1  300
C         NONE                                                          MST1  310
```

```
C                                                                               MST1  320
C.....                                                                          MST1  330
C                                                                               MST1  340
      SUBROUTINE MSET1 (ETA,CQM1,CQM2,KSET)                                     MST1  350
      IMPLICIT REAL*8(A-H,O-Z)                                                  MST1  360
      COMPLEX *16A,G,CQM1(5),CQM2(3),M4,M3,M2,M1,M0,N2,N1,N0                    MST1  370
      COMMON /COEFNT/ A,G,REY,DEL,AMDA                                          MST1  380
C                                                                               MST1  390
      AMDA = DABS(AMDA)                                                         MST1  400
      IF (KSET.EQ.0) GO TO 1                                                    MST1  410
      TETA = ETA                                                                MST1  420
      IF (KSET.EQ.1) TETA=1D0-ETA                                               MST1  430
      ETAP = AMDA*TETA                                                          MST1  440
      CNST = 1D0/DTANH(AMDA)                                                    MST1  450
      R = 1D0-CNST*DTANH(ETAP)                                                  MST1  460
      IF (KSET.EQ.-1) R = CNST*DTANH(ETAP)                                      MST1  470
C                                                                               MST1  480
      S1 = DSINH(ETAP)                                                          MST1  490
      C1 = DCOSH(ETAP)                                                          MST1  500
      T1 = DTANH(ETAP)                                                          MST1  510
C                                                                               MST1  520
      D4N = (8D0*CNST*AMDA**4)*(T1**3/C1**2)                                    MST1  530
      IF (KSET.EQ.-1) D4N=-D4N                                                  MST1  540
      D3N = (-2D0*CNST*AMDA**3)*(1D0-2D0*S1**2)/C1**4                           MST1  550
      D2N = (2D0*CNST*AMDA**2)*T1/C1**2                                         MST1  560
      IF (KSET.EQ.-1) D2N=-D2N                                                  MST1  570
      D1N = (CNST*AMDA)*(1D0/C1**2)                                             MST1  580
C                                                                               MST1  590
      F11 = 1D0/D1N                                                             MST1  600
      F22 = 1D0/D1N**2                                                          MST1  610
      F21 = (-1D0/D1N**3)*D2N                                                   MST1  620
      F33 = 1D0/D1N**3                                                          MST1  630
      F32 = (-3D0/D1N**4)*D2N                                                   MST1  640
      F31 = (3D0/D1N**5)*D2N**2-(1D0/D1N**4)*D3N                                MST1  650
      F44 = 1D0/D1N**4                                                          MST1  660
      F43 = (-6D0/D1N**5)*D2N                                                   MST1  670
      F42 = (15D0/D1N**6)*D2N**2-(4D0/D1N**5)*D3N                               MST1  680
      F41 = (-15D0/D1N**7)*D2N**3+(10D0/D1N**6)*D2N*D3N-(1D0/D1N**5)*D4N        MST1  690
C                                                                               MST1  700
    1 IF (KSET.EQ.0) R = ETA                                                    MST1  710
C                                                                               MST1  720
C        DEFINE THE RECURRING PARAMETER U(R).                                   MST1  730
C                                                                               MST1  740
C                                                                               MST1  750
      U = 2D0*(1D0-R**2)                                                        MST1  760
C                                                                               MST1  770
C                                                                               MST1  780
C        COEFFICIENTS OF THE COMPONENT Q.                                       MST1  790
```

```
C                                                                       MST1  800
C                                                                       MST1  810
      M4 = -R/REY                                                       MST1  820
      M3 = -6D0/REY                                                     MST1  830
      M2 = R*A*U-1D0/REY*(3D0/R+2D0*A**2*R)                             MST1  840
      M1 = 3D0*A*U+1D0/REY*(3D0/R**2-6D0*A**2)                          MST1  850
      M0 = R*A**3*U-1D0/REY*(A**4*R)                                    MST1  860
C                                                                       MST1  870
      N2 = -R                                                           MST1  880
      N1 = -3D0                                                         MST1  890
      N0 = -A**2*R                                                      MST1  900
C                                                                       MST1  910
      IF (KSET.EQ.0) GO TO 2                                            MST1  920
C                                                                       MST1  930
      CQM1(5) = M4*F44                                                  MST1  940
      CQM1(4) = M4*F43+M3*F33                                           MST1  950
      CQM1(3) = M4*F42+M3*F32+M2*F22                                    MST1  960
      CQM1(2) = M4*F41+M3*F31+M2*F21+M1*F11                             MST1  970
      CQM1(1) = M0                                                      MST1  980
C                                                                       MST1  990
      CQM2(3) = N2*F22                                                  MST11000
      CQM2(2) = N2*F21+N1*F11                                           MST11010
      CQM2(1) = N0                                                      MST11020
      RETURN                                                            MST11030
C                                                                       MST11040
    2 CQM1(5) = M4                                                      MST11050
      CQM1(4) = M3                                                      MST11060
      CQM1(3) = M2                                                      MST11070
      CQM1(2) = M1                                                      MST11080
      CQM1(1) = M0                                                      MST11090
C                                                                       MST11100
      CQM2(3) = N2                                                      MST11110
      CQM2(2) = N1                                                      MST11120
      CQM2(1) = N0                                                      MST11130
C                                                                       MST11140
      RETURN                                                            MST11150
      END                                                               MST11160
C
C
C.....SUBROUTINE MSET2(X,N,MDIM,CFMAT,KSET).....MST2   10
C                                                                       MST2   20
C     PURPOSE                                                           MST2   30
C                                                                       MST2   40
C       MSET2 GENERATES THE ARRAYS REPRESENTING THE CENTRAL DIFFERENCE   MST2   50
C       APPROXIMATION OF THE VORTICITY TRANSPORT EQUATION IN TERMS OF    MST2   60
C       THE VELOCITY VECTOR POTENTIAL.                                  MST2   70
C                                                                       MST2   80
C     USAGE                                                             MST2   90
```

```
C                                                                       MST2  100
C       CALL MSET2(X,N,MDIM,CFMAT,KSET)                                 MST2  110
C                                                                       MST2  120
C     DESCRIPTION OF PARAMETERS                                         MST2  130
C                                                                       MST2  140
C       X - THE NAME OF THE ARRAY BEING GENERATED. MUST BE DIMENSIONED  MST2  150
C           IN THE CALLING PROGRAM                                      MST2  160
C                                                                       MST2  170
C       N- THE ROW DIMENSION OF THE MATRIX X.  MUST BE .GE. N.          MST2  180
C                                                                       MST2  190
C       MDIM - THE COLUMN DIMENSION OF THE MATRIX X. MUST BE .GE. N.    MST2  200
C                                                                       MST2  210
C       CFMAT - THE NAME OF A FUNCTION SUBPROGRAM WITH 4 PARAMETERS,    MST2  220
C               JSTA, K,CQM1 & CQM2.  CFMAT MUST BE DECLARED            MST2  230
C               EXTERNAL IN THE CALLING PROGRAM.                        MST2  240
C                                                                       MST2  250
C     THE FOLLOWING IS OUTPUT BY MSET2                                  MST2  260
C                                                                       MST2  270
C       X - THE N BY N MATRIX INTO WHICH THE COEFFICIENTS OF THE CENTRALMST2  280
C           DIFFERENCING ARE PUT.                                       MST2  290
C                                                                       MST2  300
C     OTHER ROUTINES NEEDED                                             MST2  310
C                                                                       MST2  320
C       FUNCTION SUBPROGRAM NAME PASSED IN THE CALLING PARAMETER 'CFMAT'MST2  330
C       AND MSET1.                                                      MST2  340
C.....................................................................MST2  350
C                                                                       MST2  360
      SUBROUTINE MSET2 (X,N,MDIM,CFMAT,KSET)                            MST2  370
      REAL *8REY,R,DEL,DFLOAT,AMDA,ETA                                  MST2  380
      COMPLEX *16X(MDIM,MDIM),CQM1(5),CQM2(3)                           MST2  390
      COMPLEX *16A,G                                                    MST2  400
      COMPLEX *16CFMAT                                                  MST2  410
      COMMON /COEFNT/ A,G,REY,DEL,AMDA                                  MST2  420
C                                                                       MST2  430
C                                                                       MST2  440
C        DEFINE THE SPACING OF THE INTERIOR MESH POINTS.                MST2  450
C                                                                       MST2  460
C                                                                       MST2  470
      DEL = 1DO/DFLOAT(N+1)                                             MST2  480
C                                                                       MST2  490
C                                                                       MST2  500
C        INITIALIZE ALL ELEMENTS IN THE ARRAY TO ZERO.                  MST2  510
C                                                                       MST2  520
C                                                                       MST2  530
C                                                                       MST2  540
      DO 1 I=1,N                                                        MST2  550
C                                                                       MST2  560
      DO 1 J=1,N                                                        MST2  570
```

```
    1 X(I,J) = (0D0,0D0)                                                    MST2  580
C                                                                           MST2  590
CCC                                                                         MST2  600
CCC                                                                         MST2  610
CCC       ESTABLISH THE CENTRAL DIFFERENCE APPROXIMATION AT EACH POINT IN   MST2  620
CCC       THE MESH.                                                         MST2  630
CCC                                                                         MST2  640
C                                                                           MST2  650
      ETA = DEL                                                             MST2  660
      CALL MSET1 (ETA,CQM1,CQM2,KSET)                                       MST2  670
      X(1,1) = CFMAT(1,1,CQM1,CQM2)                                         MST2  680
      X(1,2) = CFMAT(1,2,CQM1,CQM2)                                         MST2  690
      X(1,3) = CFMAT(1,3,CQM1,CQM2)                                         MST2  700
      X(1,4) = CFMAT(1,4,CQM1,CQM2)                                         MST2  710
      X(1,5) = CFMAT(1,5,CQM1,CQM2)                                         MST2  720
C                                                                           MST2  730
      ETA = 2D0*DEL                                                         MST2  740
      CALL MSET1 (ETA,CQM1,CQM2,KSET)                                       MST2  750
      X(2,1) = CFMAT(2,1,CQM1,CQM2)                                         MST2  760
      X(2,2) = CFMAT(2,2,CQM1,CQM2)                                         MST2  770
      X(2,3) = CFMAT(2,3,CQM1,CQM2)                                         MST2  780
      X(2,4) = CFMAT(2,4,CQM1,CQM2)                                         MST2  790
      X(2,5) = CFMAT(2,5,CQM1,CQM2)                                         MST2  800
C                                                                           MST2  810
      IL = N-2                                                              MST2  820
C                                                                           MST2  830
      DO 2 I=3,IL                                                           MST2  840
      K = I-3                                                               MST2  850
      ETA = DEL*DFLOAT(I)                                                   MST2  860
      CALL MSET1 (ETA,CQM1,CQM2,KSET)                                       MST2  870
C                                                                           MST2  880
      DO 2 J=1,5                                                            MST2  890
    2 X(I,K+J) = CFMAT(3,J,CQM1,CQM2)                                       MST2  900
C                                                                           MST2  910
C                                                                           MST2  920
      ETA = 1D0-2D0*DEL                                                     MST2  930
      CALL MSET1 (ETA,CQM1,CQM2,KSET)                                       MST2  940
      X(N-1,N-3) = CFMAT(4,1,CQM1,CQM2)                                     MST2  950
      X(N-1,N-2) = CFMAT(4,2,CQM1,CQM2)                                     MST2  960
      X(N-1,N-1) = CFMAT(4,3,CQM1,CQM2)                                     MST2  970
      X(N-1,N) = CFMAT(4,4,CQM1,CQM2)                                       MST2  980
C                                                                           MST2  990
      ETA = 1D0-DEL                                                         MST21000
      CALL MSET1 (ETA,CQM1,CQM2,KSET)                                       MST21010
      X(N,N-3) = CFMAT(5,1,CQM1,CQM2)                                       MST21020
      X(N,N-2) = CFMAT(5,2,CQM1,CQM2)                                       MST21030
      X(N,N-1) = CFMAT(5,3,CQM1,CQM2)                                       MST21040
      X(N,N) = CFMAT(5,4,CQM1,CQM2)                                         MST21050
```

```
      RETURN                                                                MST21060
      END                                                                   MST21070
C
C
C.....................FUNCTION CQM1E1(JSTA,K,CQM1,CQM2).............CQM1   10
C                        (POLAR COORDINATES)                          CQM1   20
C                                                                     CQM1   30
C     PURPOSE                                                         CQM1   40
C                                                                     CQM1   50
C         RETURNS THE VALUES FOR THE COEFFICIENTS IN THE ARRAYS       CQM1   60
C         REPRESENTING THE CENTRAL DIFFERENCE APPROXIMATION OF THE    CQM1   70
C         VORTICITY TRANSPORT EQUATION USING THE COEFFICIENTS COMPUTED CQM1   80
C         BY SUBROUTINE MSET1.                                        CQM1   90
C                                                                     CQM1  100
C     DESCRIPTION OF PARAMETERS                                       CQM1  110
C                                                                     CQM1  120
C        JSTA - INDICATES WHICH DIFFERENCE EQUATION SET WILL BE USED. CQM1  130
C                                                                     CQM1  140
C             JSTA=1 - CONSISTENT 3RD ORDER TRUNCATION ERROR FINITE   CQM1  150
C                      DIFFERENCE EQUATIONS FOR R=DEL WILL BE USED.   CQM1  160
C             JSTA=2 - SAME AS ABOVE BUT R=2D0*DEL                    CQM1  170
C             JSTA=3 - CENTRAL DIFFERENCE EQUATIONS WITH CONSISTENT 2ND CQM1  180
C                      ORDER TRUNCATION ERROR WILL BE USED.           CQM1  190
C             JSTA=4 - SAME AS JSTA=3 BUT FOR R=1D0-2D0*DEL.          CQM1  200
C             JSTA=5 - SAME AS ABOVE BUT FOR R=1D0-DEL.               CQM1  210
C                                                                     CQM1  220
C                                                                     CQM1  230
C        K - INDICATES THE ABSOLUTE POSITION OF THE POINT IN EACH ROW CQM1  240
C            OF THE FINITE DIFFERENCE MESH.  IF THE FIRST NON-ZERO ENTRYCQM1  250
C            IN ROW J IS ELEMENT(J,3), THEN K=1 DENOTES ELEMENT(J,3), CQM1  260
C            K=2 DENOTES ELEMENT(J,4),ETC.                            CQM1  270
C                                                                     CQM1  280
C        CQM1,CQM2 - THE COEFFICIENT ARRAYS FOR THE FINITE DIFFERENCE CQM1  290
C             APPROXIMATION OF THE FUNCTION Q.  CQM1 CONTAINS THE     CQM1  300
C             COEFFICIENTS FOR THE NON-GAMMA TERMS, WHILE CQM2 CONTAINS CQM1  310
C             THE COEFFICIENTS OF THE GAMMA TERMS. BOTH ARRAYS MUST BE CQM1  320
C             DIMENSIONED COMPLEX*16.                                 CQM1  330
C                                                                     CQM1  340
C     EXAMPLE OF THE CALLING ARGUMENT:                                CQM1  350
C                                                                     CQM1  360
C        CQ M(1,2) E1(JSTA,K,CQM1,CQM2)                               CQM1  370
C                                                                     CQM1  380
C             CQ - Q COMPONENT OF THE VELOCITY VECTOR POTENTIAL.      CQM1  390
C                                                                     CQM1  400
C             M(1,2) - 1 REFERS TO TERMS NOT CONTAINING GAMMA AS A    CQM1  410
C                        FACTOR.                                      CQM1  420
C                      2 REFERS TO TERMS CONTAINING GAMMA AS A FACTOR. CQM1  430
C                                                                     CQM1  440
```

```
C              E1 - REFERS TC THE LINEAR COMBINATICN OF THE FIRST AND     CQM1  450
CC                  THIRD EQUATIONS RESULTING FROM EXPRESSION OF THE      CQM1  460
CC                  VORTICITY TRANSPORT EQUATION IN TERMS OF THE VELOCITYCQM1  470
CC                  VECTOR POTENTIAL.                                     CQM1  480
CC                                                                        CQM1  490
CC     USAGE                                                              CQM1  500
CC                                                                        CQM1  510
CC       CQM1E1 MUST BE DECLARED COMPLEX*16 IN THE CALLING PROGRAM.       CQM1  520
CC                                                                        CQM1  530
CC     OTHER ROUTINES REQUIRED                                            CQM1  540
CC                                                                        CQM1  550
CC       NONE                                                             CQM1  560
C.....................................................................CQM1  570
C                                                                         CQM1  580
      FUNCTION CQM1E1 (JSTA,K,CQM1,CQM2)                                  CQM1  590
      IMPLICIT COMPLEX*16(A-H,O-Z)                                        CQM1  600
      CCMMON /COEFNT/ A,G,REY,DEL,AMDA                                    CQM1  610
      COMPLEX *16CQM1(5),CQM2(3)                                          CQM1  620
      REAL *8REY,R,DEL                                                    CQM1  630
C                                                                         CQM1  640
CC                                                                        CQM1  650
      GC TO (1,7,13,19,24), JSTA                                          CQM1  660
C                                                                         CQM1  670
CCC       FINITE DIFFERENCE EQUATIONS AT ETA=DEL (NON GAMMA).             CQM1  680
CC                                                                        CQM1  690
C                                                                         CQM1  700
    1 GO TO (2,3,4,5,6), K                                                CQM1  710
    2 CCM1E1 = -0.91156462600*CQM1(5)/DEL**4+1.82516556300*CQM1(4)/DEL**CQM1  720
     13-35D0*CQM1(3)/(60D0*DEL**2)-2D0*CQM1(2)/(3D0*DEL)+CQM1(1)          CQM1  730
      GO TO 29                                                            CQM1  740
    3 CQM1E1 = 2.75024295500*CQM1(5)/DEL**4-3.25033112600*CQM1(4)/DEL** 3CQM1  750
     1+8D0*CQM1(3)/(15D0*DEL**2)+2D0*CQM1(2)/(3D0*DEL)                    CQM1  760
      GC TO 29                                                            CQM1  770
    4 CCM1E1 = -3.04373177900*CQM1(5)/DEL**4+1.66092715200*CQM1(4)/DEL**CQM1  780
     13+CQM1(3)/(20D0*DEL**2)                                             CQM1  790
      GO TO 29                                                            CQM1  800
    5 CQM1E1 = 1.42468416D0*CQM1(5)/DEL**4-.23576158900*CQM1(4)/DEL**3    CQM1  810
      GO TO 29                                                            CQM1  820
    6 CCM1E1 = -0.21963070900*CQM1(5)/DEL**4                              CQM1  830
      GO TO 29                                                            CQM1  840
C                                                                         CQM1  850
CC        FINITE DIFFERENCE EQUATIONS AT ETA=2D0*DEL (NON GAMMA).         CQM1  860
C                                                                         CQM1  870
    7 GO TO (8,9,10,11,12), K                                             CQM1  880
    8 CCM1E1 = -3.10340136D0*CQM1(5)/DEL**4+.86887417200*CQM1(4)/DEL**3+CQM1  890
     111D0*CQM1(3)/(12D0*DEL**2)-4D0*CQM1(2)/(3D0*DEL)                    CQM1  900
      GO TO 29                                                            CQM1  910
    9 CQM1E1 = 6.90301263400*CQM1(5)/DEL**4-.93774834500*CQM1(4)/DEL**3-CQM1  920
```

```
     128D0*CQM1(3)/(15D0*DEL**2)+4D0*CQM1(2)/(3D0*DEL)+CQM1(1)                  CQM1  930
      GO TO 29                                                                 CQM1  940
   10 CQM1E1 = -5.342274053D0*CQM1(5)/DEL**4-.254304636D0*CQM1(4)/DEL**3       CQM1  950
     1+19D0*CQM1(3)/(20D0*DEL**2)                                              CQM1  960
      GO TO 29                                                                 CQM1  970
   11 CQM1E1 = 1.666083577D0*CQM1(5)/DEL**4+.323178808D0*CQM1(4)/DEL**3        CQM1  980
      GO TO 29                                                                 CQM1  990
   12 CQM1E1 = -0.123420797D0*CQM1(5)/DEL**4                                   CQM11000
      GO TO 29                                                                 CQM11010
C                                                                              CQM11020
CCC      CENTRAL DIFFERENCE APPROXIMATION FOR COMPONENT Q (NON GAMMA).         CQM11030
C                                                                              CQM11040
   13 GO TO (14,15,16,17,18), K                                                CQM11050
   14 CQM1E1 = CQM1(5)/DEL**4-CQM1(4)/(2D0*DEL**3)                             CQM11060
      GO TO 29                                                                 CQM11070
   15 CQM1E1 = -4D0*CQM1(5)/DEL**4+CQM1(4)/DEL**3+CQM1(3)/DEL**2-CQM1(2)      CQM11080
     1/(2D0*DEL)                                                               CQM11090
      GO TO 29                                                                 CQM11100
   16 CQM1E1 = 6D0*CQM1(5)/DEL**4-2D0*CQM1(3)/DEL**2+CQM1(1)                   CQM11110
      GO TO 29                                                                 CQM11120
   17 CQM1E1 = -4D0*CQM1(5)/DEL**4-CQM1(4)/DEL**3+CQM1(3)/DEL**2+CQM1(2)      CQM11130
     1/(2D0*DEL)                                                               CQM11140
      GO TO 29                                                                 CQM11150
   18 CQM1E1 = CQM1(5)/DEL**4+CQM1(4)/(2D0*DEL**3)                             CQM11160
      GO TO 29                                                                 CQM11170
C                                                                              CQM11180
CCC      FINITE DIFFERENCE EQUATIONS AT ETA=1D0-2D0*DEL (NCN-GAMMA).           CQM11190
C                                                                              CQM11200
   19 GO TO (20,21,22,23), K                                                   CQM11210
   20 CQM1E1 = CQM1(5)/DEL**4-0.5D0*CQM1(4)/DEL**3                             CQM11220
      GO TO 29                                                                 CQM11230
   21 CQM1E1 = -4D0*CQM1(5)/DEL**4+CQM1(4)/DEL**3+CQM1(3)/DEL**2-CQM1(2)      CQM11240
     1/(2D0*DEL)                                                               CQM11250
      GO TO 29                                                                 CQM11260
   22 CQM1E1 = 6D0*CQM1(5)/DEL**4-2D0*CQM1(3)/DEL**2+CQM1(1)                   CQM11270
      GO TO 29                                                                 CQM11280
   23 CQM1E1 = -4D0*CQM1(5)/DEL**4-CQM1(4)/DEL**3+CQM1(3)/DEL**2+CQM1(2)      CQM11290
     1/(2D0*DEL)                                                               CQM11300
      GO TO 29                                                                 CQM11310
C                                                                              CQM11320
CCC      FINITE DIFFERENCE EQUATIONS AT ETA=1D0-DEL (NON GAMMA).               CQM11330
C                                                                              CQM11340
   24 GO TO (25,26,27,28), K                                                   CQM11350
   25 CQM1E1 = -0.25D0*CQM1(5)/DEL**4                                          CQM11360
      GO TO 29                                                                 CQM11370
   26 CQM1E1 = 8D0*CQM1(5)/(3D0*DEL**4)-CQM1(4)/(3D0*DEL**3)                   CQM11380
      GO TO 29                                                                 CQM11390
   27 CQM1E1 = -9D0*CQM1(5)/DEL**4+CQM1(3)/DEL**2-CQM1(2)/(2D0*DEL)            CQM11400
```

```
      GO TO 29                                                          CQM11410
   28 CQM1E1 = 16D0*CQM1(5)/DEL**4+3D0*CQM1(4)/DEL**3-2D0*CQM1(3)/DEL**2CQM11420
     1+CQM1(1)                                                          CQM11430
   29 RETURN                                                            CQM11440
C                                                                       CQM11450
      ENTRY CQM2E1(JSTA,K,CQM1,CQM2)                                    CQM11460
C                                                                       CQM11470
      GO TO (30,35,40,45,50), JSTA                                      CQM11480
C                                                                       CQM11490
CC       FINITE DIFFERENCE EQUATIONS AT ETA=DEL (GAMMA).                CQM11500
C                                                                       CQM11510
   30 GO TO (31,32,33,34,34), K                                         CQM11520
   31 CQM2E1 = -35D0*CQM2(3)/(60D0*DEL**2)-2D0*CQM2(2)/(3D0*DEL)+CQM2(1)CQM11530
      GO TO 54                                                          CQM11540
   32 CQM2E1 = 8D0*CQM2(3)/(15D0*DEL**2)+2D0*CQM2(2)/(3D0*DEL)          CQM11550
      GO TO 54                                                          CQM11560
   33 CQM2E1 = CQM2(3)/(20D0*DEL**2)                                    CQM11570
      GO TO 54                                                          CQM11580
   34 CQM2E1 = (0D0,0D0)                                                CQM11590
      GO TO 54                                                          CQM11600
C                                                                       CQM11610
CC       FINITE DIFFERENCE EQUATIONS AT ETA=2D0*DEL (GAMMA)             CQM11620
C                                                                       CQM11630
   35 GO TO (36,37,38,39,39), K                                         CQM11640
   36 CQM2E1 = 11D0*CQM2(3)/(12D0*DEL**2)-4D0*CQM2(2)/(3D0*DEL)         CQM11650
      GO TO 54                                                          CQM11660
   37 CQM2E1 = -28D0*CQM2(3)/(15D0*DEL**2)+4D0*CQM2(2)/(3D0*DEL)+CQM2(1)CQM11670
      GO TO 54                                                          CQM11680
   38 CQM2E1 = 19D0*CQM2(3)/(20D0*DEL**2)                               CQM11690
      GO TO 54                                                          CQM11700
   39 CQM2E1 = (0D0,0D0)                                                CQM11710
      GO TO 54                                                          CQM11720
C       CENTRAL DIFFERENCE EQUATIONS FOR THE COMPONENT Q (GAMMA).       CQM11730
C                                                                       CQM11740
   40 GO TO (41,42,43,44,41), K                                         CQM11750
   41 CQM2E1 = (0D0,0D0)                                                CQM11760
      GO TO 54                                                          CQM11770
   42 CQM2E1 = CQM2(3)/DEL**2-CQM2(2)/(2D0*DEL)                         CQM11780
      GO TO 54                                                          CQM11790
   43 CQM2E1 = -2D0*CQM2(3)/DEL**2+CQM2(1)                              CQM11800
      GO TO 54                                                          CQM11810
   44 CQM2E1 = CQM2(3)/DEL**2+CQM2(2)/(2D0*DEL)                         CQM11820
      GO TO 54                                                          CQM11830
C                                                                       CQM11840
CC       FINITE DIFFERENCE EQUATIONS AT ETA=1D0-2D0*DEL (GAMMA).        CQM11850
C                                                                       CQM11860
   45 GO TO (49,46,47,48), K                                            CQM11870
   46 CQM2E1 = CQM2(3)/DEL**2-CQM2(2)/(2D0*DEL)                         CQM11880
```

```
      GO TO 54                                                          CQM11890
   47 CQM2E1 = -2D0*CQM2(3)/DEL**2+CQM2(1)                              CQM11900
      GO TO 54                                                          CQM11910
   48 CQM2E1 = CQM2(3)/DEL**2+CQM2(2)/(2D0*DEL)                         CQM11920
      GO TO 54                                                          CQM11930
   49 CQM2E1 = (0D0,0D0)                                                CQM11940
      GO TO 54                                                          CQM11950
C                                                                       CQM11960
C        FINITE DIFFERENCE EQUATIONS AT ETA=1D0-DEL (GAMMA).            CQM11970
C                                                                       CQM11980
   50 GO TO (53,53,51,52), K                                            CQM11990
   51 CQM2E1 = CQM2(3)/DEL**2-CQM2(2)/(2D0*DEL)                         CQM12000
      GO TO 54                                                          CQM12010
   52 CQM2E1 = -2D0*CQM2(3)/DEL**2+CQM2(1)                              CQM12020
      GO TO 54                                                          CQM12030
   53 CQM2E1 = (0D0,0D0)                                                CQM12040
   54 RETURN                                                            CQM12050
      END                                                               CQM12060
CCC
CCC.....SUBROUTINE CDMTIN(N,A,NDIM,IERR).....CDMT  10
CCC                                                                     CDMT  20
CCC     PURPOSE                                                         CDMT  30
CCC                                                                     CDMT  40
CCC        INVERT A COMPLEX*16 MATRIX                                   CDMT  50
CCC                                                                     CDMT  60
CCC     USAGE                                                           CDMT  70
CCC                                                                     CDMT  80
CCC        CALL CDMTIN(N,A,NDIM,DETERM)                                 CDMT  90
CCC                                                                     CDMT 100
CCC     DESCRIPTION OF PARAMETERS                                       CDMT 110
CCC                                                                     CDMT 120
CCC        N       - ORDER OF COMPLEX*16 MATRIX TO BE INVERTED          CDMT 130
CCC                  (INTEGER) MAXIMUM 'N' IS 100                       CDMT 140
CCC                                                                     CDMT 150
CCC        A       - COMPLEX*16 INPUT MATRIX (DESTROYED).  THE          CDMT 160
CCC                  INVERSE OF 'A' IS RETURNED IN ITS PLACE            CDMT 170
CCC                                                                     CDMT 180
CCC        NDIM    - THE SIZE TO WHICH 'A' IS DIMENSIONED               CDMT 190
CCC                  (ROW DIMENSION OF 'A' ACTUALLY APPEARING           CDMT 200
CCC                  IN THE DIMENSION STATMENT OF USER'S                CDMT 210
CCC                  CALLING PROGRAM)                                   CDMT 220
CCC                                                                     CDMT 230
CCC        IERR - ERROR PARAMETER RETURNED BY CDMTIN. IERR = 0 INDICATES CDMT 240
CCC               NORMAL INVERSION. IERR = 9999 INDICATES SINGULAR MATRIX.CDMT 250
CCC                                                                     CDMT 260
CCC     REMARKS                                                         CDMT 270
C                                                                       CDMT 280
```

```
C          MATRIX 'A' MUST BE A COMPLEX*16 GENERAL MATRIX                    CDMT 290
C          IF MATRIX 'A' IS SINGULAR THAT MESSAGE IS PRINTED                 CDMT 300
C          'N' MUST BE .LE. NDIM                                             CDMT 310
C                                                                            CDMT 320
C       SUBROUTINES AND FUNCTIONS REQUIRED                                   CDMT 330
C                                                                            CDMT 340
C          ONLY BUILT-IN FORTRAN FUNCTIONS                                   CDMT 350
C                                                                            CDMT 360
C       METHOD                                                               CDMT 370
C                                                                            CDMT 380
C          GAUSSIAN ELIMINATION WITH COLUMN PIVOTING IS USED.                CDMT 390
C                                                                            CDMT 400
C.....................................................................       CDMT 410
C                                                                            CDMT 420
      SUBROUTINE CDMTIN (N,A,NDIM,IERR)                                      CDMT 430
      IMPLICIT REAL*8(A-H,O-Z)                                               CDMT 440
      INTEGER *4IPIVOT(100),INDEX(100,2),IERR                                CDMT 450
      REAL *8TEMP,ALPHA(100)                                                 CDMT 460
      COMPLEX *16A(NDIM,NDIM),PIVOT(100),AMAX,T,SWAP,U                       CDMT 470
C                                                                            CDMT 480
C       INITIALIZATION                                                       CDMT 490
C                                                                            CDMT 500
C                                                                            CDMT 510
      DO 2 J=1,N                                                             CDMT 520
      IERR = 0                                                               CDMT 530
      ALPHA(J) = 0D0                                                         CDMT 540
C                                                                            CDMT 550
      DO 1 I=1,N                                                             CDMT 560
    1 ALPHA(J) = ALPHA(J)+A(J,I)*DCONJG(A(J,I))                              CDMT 570
C                                                                            CDMT 580
      ALPHA(J) = DSQRT(ALPHA(J))                                             CDMT 590
    2 IPIVOT(J) = 0                                                          CDMT 600
C                                                                            CDMT 610
C                                                                            CDMT 620
      DO 16 I=1,N                                                            CDMT 630
C                                                                            CDMT 640
C       SEARCH FOR PIVOT ELEMENT                                             CDMT 650
C                                                                            CDMT 660
      AMAX = (0D0,0D0)                                                       CDMT 670
C                                                                            CDMT 680
      DO 7 J=1,N                                                             CDMT 690
      IF (IPIVOT(J)-1) 3,7,3                                                 CDMT 700
C                                                                            CDMT 710
    3 DO 6 K=1,N                                                             CDMT 720
      IF (IPIVOT(K)-1) 4,6,21                                                CDMT 730
    4 TEMP = AMAX*DCONJG(AMAX)-A(J,K)*DCONJG(A(J,K))                         CDMT 740
      IF (TEMP) 5,5,6                                                        CDMT 750
    5 IROW = J                                                               CDMT 760
```

```
      ICOLUM = K                                                        CDMT 770
      AMAX = A(J,K)                                                     CDMT 780
    6 CONTINUE                                                          CDMT 790
C                                                                       CDMT 800
    7 CONTINUE                                                          CDMT 810
C                                                                       CDMT 820
      IPIVOT(ICOLUM) = IPIVOT(ICOLUM)+1                                 CDMT 830
C                                                                       CDMT 840
C       INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL               CDMT 850
C                                                                       CDMT 860
      IF (IROW-ICOLUM) 8,10,8                                           CDMT 870
    8 CONTINUE                                                          CDMT 880
C                                                                       CDMT 890
      DO 9 L=1,N                                                        CDMT 900
      SWAP = A(IROW,L)                                                  CDMT 910
      A(IROW,L) = A(ICOLUM,L)                                           CDMT 920
    9 A(ICOLUM,L) = SWAP                                                CDMT 930
C                                                                       CDMT 940
      SWAP = ALPHA(IROW)                                                CDMT 950
      ALPHA(IROW) = ALPHA(ICOLUM)                                       CDMT 960
      ALPHA(ICOLUM) = SWAP                                              CDMT 970
   10 INDEX(I,1) = IROW                                                 CDMT 980
      INDEX(I,2) = ICOLUM                                               CDMT 990
      PIVOT(I) = A(ICOLUM,ICOLUM)                                       CDMT1000
      U = PIVOT(I)                                                      CDMT1010
      TEMP = PIVOT(I)*DCONJG(PIVOT(I))                                  CDMT1020
      IF (TEMP) 11,20,11                                                CDMT1030
C                                                                       CDMT1040
C       DIVIDE PIVOT ROW BY PIVOT ELEMENT                               CDMT1050
C                                                                       CDMT1060
   11 A(ICOLUM,ICOLUM) = (1.0D0,0.0D0)                                  CDMT1070
C                                                                       CDMT1080
      DO 12 L=1,N                                                       CDMT1090
      U = PIVOT(I)                                                      CDMT1100
   12 A(ICOLUM,L) = A(ICOLUM,L)/U                                       CDMT1110
C                                                                       CDMT1120
C                                                                       CDMT1130
C       REDUCE NON-PIVOT ROWS                                           CDMT1140
C                                                                       CDMT1150
C                                                                       CDMT1160
      DO 15 L1=1,N                                                      CDMT1170
      IF (L1-ICOLUM) 13,15,13                                           CDMT1180
   13 T = A(L1,ICOLUM)                                                  CDMT1190
      A(L1,ICOLUM) = (0.0D0,0.0D0)                                      CDMT1200
C                                                                       CDMT1210
      DO 14 L=1,N                                                       CDMT1220
      U = A(ICOLUM,L)                                                   CDMT1230
   14 A(L1,L) = A(L1,L)-U*T                                             CDMT1240
```

```
C                                                                       CDMT1250
   15 CONTINUE                                                          CDMT1260
C                                                                       CDMT1270
   16 CONTINUE                                                          CDMT1280
C                                                                       CDMT1290
CC                                                                      CDMT1300
CC      INTERCHANGE COLUMNS                                             CDMT1310
CC                                                                      CDMT1320
C                                                                       CDMT1330
      DO 19 I=1,N                                                       CDMT1340
      L = N+1-I                                                         CDMT1350
      IF (INDEX(L,1)-INDEX(L,2)) 17,19,17                               CDMT1360
   17 JROW = INDEX(L,1)                                                 CDMT1370
      JCOLUM = INDEX(L,2)                                               CDMT1380
C                                                                       CDMT1390
      DO 18 K=1,N                                                       CDMT1400
      SWAP = A(K,JROW)                                                  CDMT1410
      A(K,JROW) = A(K,JCOLUM)                                           CDMT1420
      A(K,JCOLUM) = SWAP                                                CDMT1430
   18 CONTINUE                                                          CDMT1440
C                                                                       CDMT1450
   19 CONTINUE                                                          CDMT1460
C                                                                       CDMT1470
      RETURN                                                            CDMT1480
   20 WRITE (6,22)                                                      CDMT1490
      IERR = 9999                                                       CDMT1500
   21 RETURN                                                            CDMT1510
C                                                                       CDMT1520
   22 FORMAT (20H  MATRIX IS SINGULAR)                                  CDMT1530
      END                                                               CDMT1540
C
CC.....SUBROUTINE MULM(X1,X2,N,MDIM,TEMPV).....MULM  10
CC                                                                      MULM  20
CC      PURPOSE                                                         MULM  30
CC                                                                      MULM  40
CC        PERFORMS THE MATRIX MULTIPLICATION OF A SQUARE MATRIX BY A    MULM  50
CC        SQUARE MATRIX.  THE RESULT IS RETURNED IN MATRIX X1.          MULM  60
CC                                                                      MULM  70
CC      USAGE                                                           MULM  80
CC                                                                      MULM  90
CC        CALL MULM(X1,X2,N,MDIM,TEMPV)                                 MULM 100
CC                                                                      MULM 110
CC      DESCRIPTION OF PARAMETERS                                       MULM 120
CC                                                                      MULM 130
CC        X1 - THE MULTIPLYING MATRIX ON INPUT AND THE RESULTANT PRODUCT MULM 140
CC             ON OUTPUT.                                               MULM 150
C                                                                       MULM 160
```

```
C       X2 - THE MULTIPLIED MATRIX.                                          MULM 170
C                                                                            MULM 180
C       N - THE ORDER OF X1 AND X2.                                          MULM 190
C                                                                            MULM 200
C       MDIM - THE DIMENSION OF X1 AND X2 FROM THE CALLING PROGRAM.          MULM 210
C                                                                            MULM 220
C       TEMPV - A WORKING VECTOR. MUST BE DIMENSIONED MDIM.                  MULM 230
C                                                                            MULM 240
C     OTHER ROUTINES REQUIRED                                                MULM 250
C                                                                            MULM 260
C       NONE                                                                 MULM 270
C*************************************************************************MULM 280
C                                                                            MULM 290
      SUBROUTINE MULM (X1,X2,N,MDIM,TEMPV)                                   MULM 300
      COMPLEX *16X1(MDIM,MDIM),X2(MDIM,MDIM),TEMPV(MDIM),TEMP                MULM 310
C                                                                            MULM 320
C                                                                            MULM 330
C        STORE ROW I OF X1 IN TEMPV.                                         MULM 340
C                                                                            MULM 350
C                                                                            MULM 360
C                                                                            MULM 370
      DO 4 I=1,N                                                             MULM 380
C                                                                            MULM 390
      DO 1 J=1,N                                                             MULM 400
    1 TEMPV(J) = X1(I,J)                                                     MULM 410
C                                                                            MULM 420
C                                                                            MULM 430
C                                                                            MULM 440
C        MULTIPLY COLUMN J OF X2 BY ROW I OF X1 AND STORE IN X1(I,J).        MULM 450
C                                                                            MULM 460
C                                                                            MULM 470
C                                                                            MULM 480
      DO 3 J=1,N                                                             MULM 490
      TEMP = (0D0,0D0)                                                       MULM 500
C                                                                            MULM 510
C                                                                            MULM 520
      DO 2 K=1,N                                                             MULM 530
    2 TEMP = TEMP+TEMPV(K)*X2(K,J)                                           MULM 540
C                                                                            MULM 550
C                                                                            MULM 560
    3 X1(I,J) = TEMP                                                         MULM 570
C                                                                            MULM 580
    4 CONTINUE                                                               MULM 590
C                                                                            MULM 600
      RETURN                                                                 MULM 610
      END                                                                    MULM 620
C
C
```

```
C.....SUBROUTINE DSPLIT(N,MDIM,A,AR,AI).....DSPL  10
C                                           DSPL  20
C                                           DSPL  30
C     PURPOSE                               DSPL  40
C                                           DSPL  50
C       DSPLIT TAKES A MATRIX OF COMPLEX*16 NUMBERS AND  DSPL  60
C       SPLITS IT INTO TWO MATRICES, ONE CONTAINING THE REAL  DSPL  70
C       PART OF THE ORIGINAL MATRIX, AND ONE CONTAINING THE  DSPL  80
C       IMAGINARY PART.                     DSPL  90
C                                           DSPL 100
C     USAGE                                 DSPL 110
C                                           DSPL 120
C       CALL DSPLIT(N,MDIM,A,AREAL,AIMAG)   DSPL 130
C                                           DSPL 140
C     DESCRIPTION OF PARAMETERS             DSPL 150
C                                           DSPL 160
C       N - THE SIZE OF THE MATRIX A, AN N BY N SQUARE  DSPL 170
C         MATRIX.                           DSPL 180
C                                           DSPL 190
C       MDIM - THE COLUMN DIMENSION OF MATRIX A  DSPL 200
C                                           DSPL 210
C       A - THE INPUT MATRIX.  MUST BE DIMENSIONED MDIM BY  DSPL 220
C         AT LEAST N IN THE CALLING PROGRAM  (COMPLEX*16)  DSPL 230
C                                           DSPL 240
C       AREAL,AIMAG - THE OUTPUT MATRICES CONTAINING THE  DSPL 250
C         REAL AND IMAGINARY PARTS, RESPECTIVELY, OF  DSPL 260
C         MATRIX A.  MUST BE DIMENSIONED (MDIM,MDIM) IN THE  DSPL 270
C         CALLING PROGRAM.                  DSPL 280
C                                           DSPL 290
C     NOTES...                              DSPL 300
C                                           DSPL 310
C       MATRIX A AND MATRIX AREAL MAY OVERLAP IF THEY ARE  DSPL 320
C         DIMENSIONED IN THE CALLING PROGRAM AS FOLLOWS...  DSPL 330
C                                           DSPL 340
C             COMPLEX*16 A(MDIM,MDIM)       DSPL 350
C             REAL*8 AREAL(MDIM,MDIM),AIMAG(MDIM,MDIM)  DSPL 360
C             EQUIVALENCE(A(1,1),AREAL(1,1))  DSPL 370
C                                           DSPL 380
C     OTHER ROUTINES NEEDED                 DSPL 390
C                                           DSPL 400
C       NONE                                DSPL 410
C                                           DSPL 420
C                                           DSPL 430
C.....DSPL 440
C                                           DSPL 450
      SUBROUTINE DSPLIT (N,MDIM,A,AR,AI)    DSPL 460
      REAL *8A(2,MDIM,MDIM),AR(MDIM,MDIM),AI(MDIM,MDIM)  DSPL 470
C                                           DSPL 480
```

```
C                                                                       DSPL  490
      DO 1 J=1,N                                                        DSPL  500
C                                                                       DSPL  510
      DO 1 I=1,N                                                        DSPL  520
      AR(I,J) = A(1,I,J)                                                DSPL  530
    1 AI(I,J) = A(2,I,J)                                                DSPL  540
C                                                                       DSPL  550
C                                                                       DSPL  560
      RETURN                                                            DSPL  570
      END                                                               DSPL  580
C
CCC   SUBROUTINE EBALAC (AR,AI,N,IA,K,L,D)                              EBAC0010
CCC                                                                     EBAC0020
C-EBALAC--------D--------LIBRARY 1-------------------------------------EBAC0030
CCC                                                                     EBAC0040
CCC   FUNCTION              - BALANCES A COMPLEX GENERAL MATRIX AND ISOLATESEBAC0050
CCC                           EIGENVALUES WHENEVER POSSIBLE.            EBAC0060
CCC   USAGE                 - CALL EBALAC (AR,AI,N,IA,K,L,D)            EBAC0070
CCC   PARAMETERS    AR      - INPUT/OUTPUT MATRICES OF DIMENSION N BY N. EBAC0080
CCC                 AI        ON INPUT, AR AND AND AI CONTAIN THE REAL  EBAC0090
CCC                           AND IMAGINARY PARTS, RESPECTIVELY, OF     EBAC0100
CCC                           THE COMPLEX MATRIX OF ORDER N TO BE       EBAC0110
CCC                           BALANCED. ON OUTPUT, AR AND AI CONTAIN THE EBAC0120
CCC                           REAL AND IMAGINARY PARTS OF THE           EBAC0130
CCC                           TRANSFORMED MATRIX.                       EBAC0140
CCC                 N       - INPUT VARIABLE CONTAINING THE ORDER       EBAC0150
CCC                           OF THE MATRIX A = (AR,AI) TO BE BALANCED. EBAC0160
CCC                 IA      - INPUT VARIABLE CONTAINING THE ROW DIMENSION OFEBAC0170
CCC                           AR AND AI IN THE CALLING PROGRAM.         EBAC0180
CCC                 K       - OUTPUT INTEGERS CONTAINING THE BOUNDARY   EBAC0190
CCC                 L           INDICES FOR THE BALANCED MATRIX A = (AR,AI)EBAC0200
CCC                           SUCH THAT                                 EBAC0210
CCC                             AR(I,J) = 0. AND AI(I,J) = 0. IF        EBAC0220
CCC                             (1) I IS GREATER THAN J AND             EBAC0230
CCC                             (2) J = 1,...,K-1 OR                    EBAC0240
CCC                                 I = L+1,...,N                       EBAC0250
CCC                 D       - OUTPUT VECTOR OF LENGTH N CONTAINING      EBAC0260
CCC                           INFORMATION DETERMINING THE PERMUTATIONS  EBAC0270
CCC                           USED AND THE SCALING FACTORS.             EBAC0280
CCC   PRECISION             - SINGLE/DOUBLE                             EBAC0290
CCC   LANGUAGE              - FORTRAN                                   EBAC0300
C-------------------------------------------------------------------    EBAC0310
CCC   LATEST REVISION       - MARCH 9, 1977                             EBAC0320
CCC                                                                     EBAC0330
      SUBROUTINE EBALAC (AR,AI,N,IA,K,L,D)                              EBAC0340
C                                                                       EBAC0350
      DIMENSION           AR(IA,1),AI(IA,1),D(N)                        EBAC0360
```

```
      LOGICAL             NOCONV                                          EBAC0370
C                                     RADIX IS A MACHINE DEPENDENT        EBAC0380
CC                                      PARAMETER SPECIFYING THE BASE OF  EBAC0390
CC                                      THE MACHINE FLOATING POINT REPRE- EBAC0400
C                                       SENTATION                         EBAC0410
      DOUBLE PRECISION    AR,AI,D,RADIX,ZERO,ONE,PT95,B2,F,C,G,R,S        EBAC0420
      DOUBLE PRECISION    RRADIX,RB2                                      EBAC0430
      DATA                RADIX/16.0D0/                                   EBAC0440
      DATA                ZERO,ONE,PT95/0.0D0,1.0D0,0.95D0/               EBAC0450
      B2 = RADIX*RADIX                                                    EBAC0480
      RRADIX = ONE/RADIX                                                  EBAC0490
      RB2 = RRADIX*RRADIX                                                 EBAC0500
      K = 1                                                               EBAC0510
      L = N                                                               EBAC0520
      GO TO 30                                                            EBAC0530
C                                     IN-LINE PROCEDURE FOR ROW AND COLUMN EBAC0540
C                                       EXCHANGE                          EBAC0550
    5 D(M) = J                                                            EBAC0560
      IF (J .EQ. M) GO TO 20                                              EBAC0570
      DO 10 I = 1,L                                                       EBAC0580
         F = AR(I,J)                                                      EBAC0590
         AR(I,J) = AR(I,M)                                                EBAC0600
         AR(I,M) = F                                                      EBAC0610
         F = AI(I,J)                                                      EBAC0620
         AI(I,J) = AI(I,M)                                                EBAC0630
         AI(I,M) = F                                                      EBAC0640
   10 CONTINUE                                                            EBAC0650
      DO 15 I=K,N                                                         EBAC0660
         F = AR(J,I)                                                      EBAC0670
         AR(J,I) = AR(M,I)                                                EBAC0680
         AR(M,I) = F                                                      EBAC0690
         F = AI(J,I)                                                      EBAC0700
         AI(J,I) = AI(M,I)                                                EBAC0710
         AI(M,I) = F                                                      EBAC0720
   15 CONTINUE                                                            EBAC0730
   20 GO TO (25,45), IEXC                                                 EBAC0740
C                                     SEARCH FOR ROWS ISOLATING AN        EBAC0750
C                                       EIGENVALUE AND PUSH THEM DOWN     EBAC0760
   25 IF (L .EQ. 1) GO TO 115                                             EBAC0770
      L = L-1                                                             EBAC0780
C                                     DO J=L,1,-1                         EBAC0790
   30 L1 = L+1                                                            EBAC0800
      DO 40 JJ = 1,L                                                      EBAC0810
         J = L1-JJ                                                        EBAC0820
         DO 35 I = 1,L                                                    EBAC0830
            IF (I .EQ. J) GO TO 35                                        EBAC0840
            IF (AR(J,I) .NE. ZERO .OR. AI(J,I) .NE. ZERO) GO TO 40        EBAC0850
   35    CONTINUE                                                         EBAC0860
```

```
            M = L                                                          EBAC0870
            IEXC = 1                                                       EBAC0880
            GO TO 5                                                        EBAC0890
        40 CONTINUE                                                        EBAC0900
           GO TO 50                                                        EBAC0910
        C                                     SEARCH FOR COLUMNS ISOLATING AN  EBAC0920
        C                                       EIGENVALUE AND PUSH THEM LEFT  EBAC0930
        45 K = K+1                                                         EBAC0940
        50 DO 60 J = K,L                                                   EBAC0950
              DO 55 I = K,L                                                EBAC0960
                 IF (I .EQ. J) GO TO 55                                    EBAC0970
                 IF (AR(I,J) .NE. ZERO .OR. AI(I,J) .NE. ZERO) GO TO 60    EBAC0980
        55    CONTINUE                                                     EBAC0990
              M = K                                                        EBAC1000
              IEXC = 2                                                     EBAC1010
              GO TO 5                                                      EBAC1020
        60 CONTINUE                                                        EBAC1030
        C                                     BALANCE THE SUBMATRIX IN ROWS  EBAC1040
        C                                       K TO L                     EBAC1050
           DO 65 I = K,L                                                   EBAC1060
           D(I) = ONE                                                      EBAC1070
        65 CONTINUE                                                        EBAC1080
        C                                     ITERATIVE LOOP FOR NORM REDUCTION  EBAC1090
        70 NOCONV = .FALSE.                                                EBAC1100
           DO 110 I = K,L                                                  EBAC1110
              C = ZERO                                                     EBAC1120
              R = ZERO                                                     EBAC1130
              DO 75 J = K,L                                                EBAC1140
                 IF (J .EQ. I) GO TO 75                                    EBAC1150
                 C = C+DABS(AR(J,I))+DABS(AI(J,I))                         EBAC1160
                 R = R+DABS(AR(I,J))+DABS(AI(I,J))                         EBAC1180
        75    CONTINUE                                                     EBAC1200
              G = R*RRADIX                                                 EBAC1210
              F = ONE                                                      EBAC1220
              S = C+R                                                      EBAC1230
        80    IF (C .GE. G) GO TO 85                                       EBAC1240
              F = F*RADIX                                                  EBAC1250
              C = C*B2                                                     EBAC1260
              GO TO 80                                                     EBAC1270
        85    G = R*RADIX                                                  EBAC1280
        90    IF (C .LT. G) GO TO 95                                       EBAC1290
              F = F*RRADIX                                                 EBAC1300
              C = C*RB2                                                    EBAC1310
              GO TO 90                                                     EBAC1320
        C                                     BALANCE                      EBAC1330
        95    IF ((C+R)/F .GE. PT95*S) GO TO 110                           EBAC1340
              G = ONE/F                                                    EBAC1350
              D(I) = D(I)*F                                                EBAC1360
```

```
            NOCONV = .TRUE.                                                 EBAC1370
            DO 100 J = K,N                                                  EBAC1380
               AR(I,J) = AR(I,J)*G                                          EBAC1390
               AI(I,J) = AI(I,J)*G                                          EBAC1400
  100       CONTINUE                                                        EBAC1410
            DO 105 J = 1,L                                                  EBAC1420
               AR(J,I) = AR(J,I)*F                                          EBAC1430
               AI(J,I) = AI(J,I)*F                                          EBAC1440
  105       CONTINUE                                                        EBAC1450
  110    CONTINUE                                                           EBAC1460
         IF (NOCONV) GO TO 70                                               EBAC1470
  115    RETURN                                                             EBAC1480
         END                                                                EBAC1490
C
C        SUBROUTINE EHESSC (AR,AI,K,L,N,IA,ID)                              EHEC0010
C                                                                           EHEC0020
C-EHESSC--------D-------LIBRARY 1-----------------------------------------EHEC0030
C                                                                           EHEC0040
C     FUNCTION              - REDUCTION OF A COMPLEX MATRIX TO COMPLEX      EHEC0050
C                             UPPER HESSENBERG FORM.                        EHEC0060
C     USAGE                 - CALL EHESSC(AR,AI,K,L,N,IA,ID)              EHEC0070
C     PARAMETERS    AR      - INPUT/OUTPUT MATRIX OF DIMENSION N BY N.      EHEC0080
C                             ON INPUT CONTAINS THE REAL COMPONENTS OF      EHEC0090
C                             THE MATRIX TO BE REDUCED.                     EHEC0100
C                             ON OUTPUT CONTAINS THE REAL COMPONENTS        EHEC0110
C                             OF THE REDUCED HESSENBERG FORM IN THE         EHEC0120
C                             UPPER TRIANGULAR PORTION (INCLUDING MAIN      EHEC0130
C                             AND SUB-DIAGONAL) AND THE DETAILS OF          EHEC0140
C                             THE REDUCTION IN THE LOWER TRIANGULAR         EHEC0150
C                             PORTION.                                      EHEC0160
C                     AI      - INPUT/OUTPUT MATRIX OF DIMENSION N BY N     EHEC0170
C                             CONTAINING THE IMAGINARY COUNTERPARTS         EHEC0180
C                             TO AR, ABOVE.                                 EHEC0190
C                     K       - INPUT SCALAR CONTAINING THE ROW AND COLUMN   EHEC0200
C                             INDEX OF THE STARTING ELEMENT TO BE           EHEC0210
C                             REDUCED BY ROW SCALING.  FOR UNBALANCED       EHEC0220
C                             MATRICES SET K = 1.                           EHEC0230
C                     L       - INPUT SCALAR CONTAINING THE ROW AND         EHEC0240
C                             COLUMN INDEX OF THE LAST ELEMENT TO BE        EHEC0250
C                             REDUCED BY ROW SCALING. FOR UNBALANCED        EHEC0260
C                             MATRICES SET L = N.                           EHEC0270
C                     N       - INPUT SCALAR CONTAINING THE ORDER OF        EHEC0280
C                             THE MATRIX TO BE REDUCED.                     EHEC0290
C                     IA      - INPUT SCALAR CONTAINING ROW DIMENSION       EHEC0300
C                             OF AR AND AI IN THE CALLING PROGRAM.          EHEC0310
C                     ID      - OUTPUT VECTOR OF LENGTH L CONTAINING        EHEC0320
C                             DETAILS OF THE TRANSFORMATIONS.               EHEC0330
C     PRECISION             - SINGLE/DOUBLE                                 EHEC0340
```

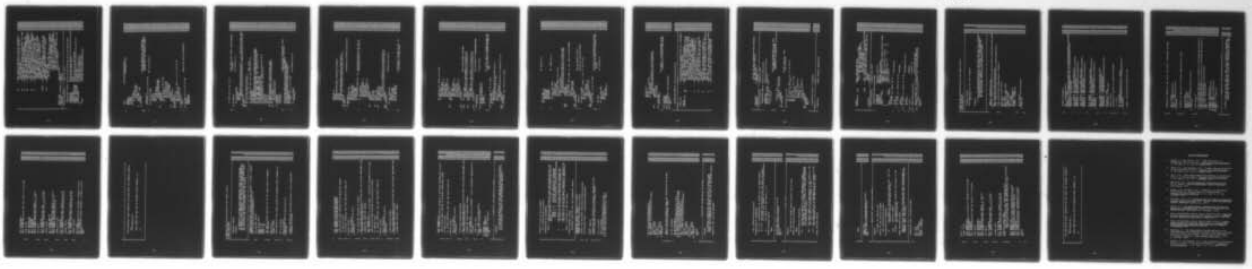
```
C   LANGUAGE              - FORTRAN                                                  EHEC0350
C-------------------------------------------------------------------------------EHEC0360
C   LATEST REVISION       - FEBRUARY 7, 1973                                         EHEC0370
C                                                                                  EHEC0380
      SUBROUTINE EHESSC (AR,AI,K,L,N,IA,ID)                                        EHEC0390
C                                                                                  EHEC0400
      DIMENSION          AR(IA,1),AI(IA,1),ID(1),T1(2),T2(2)                       EHEC0410
      DOUBLE PRECISION   AR,AI,XR,XI,YR,YI,T1,T2,ZERO                              EHEC0420
      COMPLEX*16         X,Y                                                       EHEC0430
      EQUIVALENCE        (X,T1(1),XR),(T1(2),XI),(Y,T2(1),YR),                     EHEC0450
     1                   (T2(2),YI)                                                EHEC0460
      DATA               ZERO/0.0D0/                                               EHEC0470
      LA=L-1                                                                       EHEC0490
      KP1=K+1                                                                      EHEC0500
      IF (LA .LT. KP1) GO TO 45                                                    EHEC0510
      DO 40 M=KP1,LA                                                               EHEC0520
         I=M                                                                       EHEC0530
         XR=ZERO                                                                   EHEC0540
         XI=ZERO                                                                   EHEC0550
         DO 5 J=M,L                                                                EHEC0560
            IF (DABS(AR(J,M-1))+DABS(AI(J,M-1)) .LE. DABS(XR)+DABS(XI))            EHEC0570
     1      GO TO 5                                                                EHEC0580
            XR=AR(J,M-1)                                                           EHEC0610
            XI=AI(J,M-1)                                                           EHEC0620
            I=J                                                                    EHEC0630
    5    CONTINUE                                                                  EHEC0640
         ID(M)=I                                                                   EHEC0650
         IF (I .EQ. M) GO TO 20                                                    EHEC0660
C                                        INTERCHANGE ROWS AND COLUMNS OF           EHEC0670
C                                          ARRAYS AR AND AI                        EHEC0680
         MM1=M-1                                                                   EHEC0690
         DO 10 J=MM1,N                                                             EHEC0700
            YR=AR(I,J)                                                             EHEC0710
            AR(I,J)=AR(M,J)                                                        EHEC0720
            AR(M,J)=YR                                                             EHEC0730
            YI=AI(I,J)                                                             EHEC0740
            AI(I,J)=AI(M,J)                                                        EHEC0750
            AI(M,J)=YI                                                             EHEC0760
   10    CONTINUE                                                                  EHEC0770
         DO 15 J=1,L                                                               EHEC0780
            YR=AR(J,I)                                                             EHEC0790
            AR(J,I)=AR(J,M)                                                        EHEC0800
            AR(J,M)=YR                                                             EHEC0810
            YI=AI(J,I)                                                             EHEC0820
            AI(J,I)=AI(J,M)                                                        EHEC0830
            AI(J,M)=YI                                                             EHEC0840
   15    CONTINUE                                                                  EHEC0850
C                                        END INTERCHANGE                           EHEC0860
```

```
   20     IF (XR .EQ. ZERO .AND. XI .EQ. ZERO) GO TO 40                    EHEC0870
          MP1=M+1                                                          EHEC0880
          DO 35 I=MP1,L                                                    EHEC0890
             YR=AR(I,M-1)                                                  EHEC0900
             YI=AI(I,M-1)                                                  EHEC0910
             IF (YR .EQ. ZERO .AND. YI .EQ. ZERO) GO TO 35                 EHEC0920
             Y=Y/X                                                         EHEC0930
             AR(I,M-1)=YR                                                  EHEC0940
                AR(I,J)=AR(I,J)-YR*AR(M,J)+YI*AI(M,J)                      EHEC0970
             AI(I,M-1)=YI                                                  EHEC0950
                AI(I,J)=AI(I,J)-YR*AI(M,J)-YI*AR(M,J)                      EHEC0980
             DO 25 J=M,N                                                   EHEC0960
   25        CONTINUE                                                      EHEC0990
             DO 30 J=1,L                                                   EHEC1000
                AR(J,M)=AR(J,M)+YR*AR(J,I)-YI*AI(J,I)                      EHEC1010
                AI(J,M)=AI(J,M)+YR*AI(J,I)+YI*AR(J,I)                      EHEC1020
   30        CONTINUE                                                      EHEC1030
   35     CONTINUE                                                         EHEC1040
   40  CONTINUE                                                            EHEC1050
   45  RETURN                                                              EHEC1060
       END                                                                 EHEC1070
C
C
C      SUBROUTINE ELRH2C (HR,HI,K,L,N,IH,WR,WI,ZR,ZI,ID,INFER,IER)         ELR20010
C                                                                          ELR20020
C-ELRH2C--------D-------LIBRARY 1-----------------------------------------ELR20030
C                                                                          ELR20040
C    FUNCTION              - COMPUTE THE EIGENVALUES AND EIGENVECTORS OF   ELR20050
C                            A COMPLEX UPPER HESSENBERG MATRIX AND         ELR20060
C                            BACK TRANSFORM THE EIGENVECTORS.              ELR20070
C    USAGE                 - CALL ELRH2C (HR,HI,K,L,N,IH,WR,WI,ZR,ZI,ID,  ELR20080
C                              INFER,IER)                                  ELR20090
C    PARAMETERS    HR      - INPUT MATRIX OF DIMENSION N BY N CONTAINING   ELR20100
C                            THE REAL COMPONENTS OF THE COMPLEX            ELR20110
C                            HESSENBERG MATRIX. HR IS DESTROYED ON         ELR20120
C                            OUTPUT.                                       ELR20130
C                  HI      - INPUT MATRIX OF DIMENSION N BY N CONTAINING   ELR20140
C                            THE IMAGINARY COUNTERPARTS TO HR, ABOVE.      ELR20150
C                            HI IS DESTROYED ON OUTPUT.                    ELR20160
C                  K       - INPUT SCALAR CONTAINING THE LOWER BOUNDARY    ELR20170
C                            INDEX FOR THE INPUT MATRIX.                   ELR20180
C                            FOR UNBALANCED MATRICES SET K = 1.            ELR20190
C                  L       - INPUT SCALAR CONTAINING THE UPPER BOUNDARY    ELR20200
C                            INDEX FOR THE INPUT MATRIX.                   ELR20210
C                            FOR UNBALANCED MATRICES SET L = N.            ELR20220
C                  N       - INPUT SCALAR CONTAINING THE ORDER OF THE      ELR20230
C                            HESSENBERG MATRIX AND THE EIGENVECTOR         ELR20240
C                            MATRIX.                                       ELR20250
```

AD-A066 374 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 20/4
INVESTIGATION OF PIPE FLOW INSTABILITY AND RESULTS FOR WAVE NUM--ETC(U)
DEC 78 M J ARNOLD

UNCLASSIFIED NL

2 OF 2
AD
AD66374



END
DATE
FILMED
5--79
DDC

```
C                  IH      - INPUT SCALAR CONTAINING THE ROW DIMENSION      ELR20260
C                            OF MATRICES HR, HI, ZR AND ZI IN THE           ELR20270
C                            CALLING PROGRAM.                               ELR20280
C                  WR      - OUTPUT VECTOR OF LENGTH N CONTAINING THE REAL  ELR20290
C                            COMPONENTS OF THE EIGENVALUES.                 ELR20300
C                  WI      - OUTPUT VECTOR OF LENGTH N CONTAINING THE       ELR20310
C                            IMAGINARY COMPONENTS OF THE EIGENVALUES.       ELR20320
C                  ZR      - OUTPUT MATRIX OF DIMENSION N BY N CONTAINING   ELR20330
C                            THE REAL COMPONENTS OF THE EIGENVECTORS.       ELR20340
C                            THE EIGENVECTORS ARE NOT NORMALIZED.           ELR20350
C                  ZI      - OUTPUT MATRIX OF DIMENSION N BY N CONTAINING   ELR20360
C                            THE IMAGINARY COUNTERPARTS TO ZR, ABOVE.       ELR20370
C                  ID      - INPUT VECTOR OF LENGTH N CONTAINING THE        ELR20380
C                            INFORMATION GENERATED BY IMSL ROUTINE          ELR20390
C                            EHESSC IDENTIFYING THE ROWS AND COLUMNS        ELR20400
C                            INTERCHANGED DURING THE REDUCTION TO           ELR20410
C                            HESSENBERG FORM. ONLY COMPONENTS K             ELR20420
C                            THROUGH L ARE USED.                            ELR20430
C                  INFER   - OUTPUT SCALAR CONTAINING THE INDEX OF THE      ELR20440
C                            EIGENVALUE WHICH GENERATED THE TERMINAL        ELR20450
C                            ERROR(SEE DESCRIPTION OF IER, BELOW).          ELR20460
C                  IER     - ERROR PARAMETER                                ELR20470
C                            TERMINAL ERROR = 128 + N                       ELR20480
C                              N = 1 INDICATES THE EIGENVALUE RECORDED      ELR20490
C                                    IN THE OUTPUT PARAMETER, INFER,        ELR20500
C                                    COULD NOT BE DETERMINED AFTER 30       ELR20510
C                                    ITERATIONS. IF THE J-TH EIGENVALUE     ELR20520
C                                    COULD NOT BE SO DETERMINED,            ELR20530
C                                    THEN THE EIGENVALUES J+1,J+2,...,N     ELR20540
C                                    SHOULD BE CORRECT.                     ELR20550
C     PRECISION             - SINGLE/DOUBLE                                 ELR20560
C     REQD. IMSL ROUTINES   - UERTST                                        ELR20570
C     LANGUAGE              - FORTRAN                                       ELR20580
C-----------------------------------------------------------------------    ELR20590
C     LATEST REVISION       - APRIL 5, 1977                                 ELR20600
C                                                                           ELR20610
      SUBROUTINE ELRH2C (HR,HI,K,L,N,IH,WR,WI,ZR,ZI,ID,INFER,IER)           ELR20620
C                                                                           ELR20630
      DIMENSION          HR(IH,1),HI(IH,1),WR(1),WI(1),ZR(IH,1),ZI(IH,1)  ELR20640
      DIMENSION          T1(2),T2(2),T3(2),ID(1)                          ELR20650
      COMPLEX*16         X,Y,Z                                            ELR20660
      DOUBLE PRECISION   HR,HI,WR,WI,ZR,ZI,EPS,FNORM                      ELR20680
      DOUBLE PRECISION   T1,T2,T3,XR,XI,YR,YI,ZZR,ZZI,SR,SI,TR,TI         ELR20690
      DOUBLE PRECISION   ZERO,ONE,TWO                                     ELR20700
      EQUIVALENCE        (X,T1(1),XR),(T1(2),XI),                         ELR20710
     1                   (Y,T2(1),YR),(T2(2),YI),                         ELR20720
     2                   (Z,T3(1),ZZR),(T3(2),ZZI)                        ELR20730
      DATA               ZERO,ONE,TWO/0.0D0,1.0D0,2.0D0/                  ELR20740
```

```
      DATA                    EPS/Z3410000000000000/                    ELR20750
C                                     INITIALIZE IER                    ELR20770
      IER=0                                                             ELR20780
      INFER=0                                                           ELR20790
      TR=ZERO                                                           ELR20800
      TI=ZERO                                                           ELR20810
      DC 5 I=1,N                                                        ELR20820
         DO 3 J=1,N                                                     ELR20830
            ZR(I,J)=ZERO                                                ELR20840
            ZI(I,J)=ZERO                                                ELR20850
    3    CONTINUE                                                       ELR20860
         ZR(I,I) = ONE                                                  ELR20870
    5 CONTINUE                                                          ELR20880
C                                  FORM THE MATRIX OF ACCUMULATED       ELR20890
C                                    TRANSFORMATIONS FROM THE INFOR-    ELR20900
C                                    MATION LEFT BY ROUTINE 'EHESSC'    ELR20910
      IEND=L-K-1                                                        ELR20920
      IF (IEND .LE. 0) GO TO 25                                         ELR20930
C                                  DO I=L-1,K+1,-1                      ELR20940
      DO 20 II=1,IEND                                                   ELR20950
         I=L-II                                                         ELR20960
         IP1=I+1                                                        ELR20970
         IM1=I-1                                                        ELR20980
         DO 10 M=IP1,L                                                  ELR20990
            ZR(M,I)=HR(M,IM1)                                           ELR21000
            ZI(M,I)=HI(M,IM1)                                           ELR21010
   10    CONTINUE                                                       ELR21020
         J=ID(I)                                                        ELR21030
         IF (I .EC. J) GO TO 20                                         ELR21040
         DO 15 M=I,L                                                    ELR21050
            ZR(I,M)=ZR(J,M)                                             ELR21060
            ZI(I,M)=ZI(J,M)                                             ELR21070
            ZR(J,M)=ZERO                                                ELR21080
            ZI(J,M)=ZERO                                                ELR21090
   15    CONTINUE                                                       ELR21100
         ZR(J,I)=ONE                                                    ELR21110
   20 CONTINUE                                                          ELR21120
   25 DO 30 I=1,N                                                       ELR21130
         IF (I .GE. K .AND. I .LE. L) GO TO 30                          ELR21140
         WR(I)=HR(I,I)                                                  ELR21150
         WI(I)=HI(I,I)                                                  ELR21160
   30 CONTINUE                                                          ELR21170
      NN=L                                                              ELR21180
C                                  SEARCH FOR NEXT EIGENVALUE           ELR21190
   35 IF (NN .LT. K) GO TO 150                                          ELR21200
      ITS=0                                                             ELR21210
      NNM1=NN-1                                                         ELR21220
      NNM2=NN-2                                                         ELR21230
```

```
      IF (NN .EQ. K) GO TO 50                                                ELR21240
C                                          LOOK FOR SINGLE SMALL SUB-DIAGONAL ELR21250
C                                            ELEMENT                        ELR21260
   40 NPL=NN+K                                                               ELR21270
C                                          DO M=NN,K+1,-1                   ELR21280
      DO 45 KK=K,NNM1                                                        ELR21290
         M=NPL-KK                                                            ELR21300
         MM1=M-1                                                             ELR21310
         IF (DABS(HR(M,MM1))+DABS(HI(M,MM1)) .LE. EPS*(DABS(HR(MM1,MM1))ELR21320
     1      +DABS(HI(MM1,MM1))+DABS(HR(M,M))+DABS(HI(M,M)))) GO TO 55       ELR21330
   45 CONTINUE                                                               ELR21360
   50 M=K                                                                    ELR21370
   55 IF (M .EQ. NN) GO TO 145                                               ELR21380
      IF (ITS .EQ. 30) GO TO 205                                             ELR21390
C                                          FORM SHIFT                       ELR21400
      IF (ITS .EQ. 10 .OR. ITS .EQ. 20) GO TO 60                             ELR21410
      SR=HR(NN,NN)                                                           ELR21420
      SI=HI(NN,NN)                                                           ELR21430
      XR=HR(NNM1,NN)*HR(NN,NNM1)-HI(NNM1,NN)*HI(NN,NNM1)                     ELR21440
      XI=HR(NNM1,NN)*HI(NN,NNM1)+HI(NNM1,NN)*HR(NN,NNM1)                     ELR21450
      IF (XR .EQ. ZERO .AND. XI .EQ. ZERO) GO TO 65                          ELR21460
      YR=(HR(NNM1,NNM1)-SR)/TWO                                              ELR21470
      YI=(HI(NNM1,NNM1)-SI)/TWO                                              ELR21480
      Z=CDSQRT(DCMPLX(YR**2-YI**2+XR,TWO*YR*YI+XI))                          ELR21490
      IF (YR*ZZR+YI*ZZI .LT. ZERO) Z=-Z                                      ELR21510
      X=X/(Y+Z)                                                              ELR21520
      SR=SR-XR                                                               ELR21530
      SI=SI-XI                                                               ELR21540
      GO TO 65                                                               ELR21550
   60 SR=DABS(HR(NN,NNM1))+DABS(HR(NNM1,NNM2))                               ELR21560
      SI=DABS(HI(NN,NNM1))+DABS(HI(NNM1,NNM2))                               ELR21570
   65 DO 70 I=K,NN                                                           ELR21600
         HR(I,I)=HR(I,I)-SR                                                  ELR21610
         HI(I,I)=HI(I,I)-SI                                                  ELR21620
   70 CONTINUE                                                               ELR21630
      TR=TR+SR                                                               ELR21640
      TI=TI+SI                                                               ELR21650
      ITS=ITS+1                                                              ELR21660
C                                          LOOK FOR TWO CONSECUTIVE SMALL   ELR21670
C                                            SUB-DIAGONAL ELEMENTS          ELR21680
      XR=DABS(HR(NNM1,NNM1))+DABS(HI(NNM1,NNM1))                             ELR21690
      YR=DABS(HR(NN,NNM1))+DABS(HI(NN,NNM1))                                 ELR21700
      ZZR=DABS(HR(NN,NN))+DABS(HI(NN,NN))                                    ELR21710
      NNMJ=NNM1-M                                                            ELR21750
      IF (NNMJ .EQ. 0) GO TO 80                                              ELR21760
C                                          DO MM=NN-1,M+1,-1                ELR21770
      DO 75 NM=1,NNMJ                                                        ELR21780
         MM=NN-NM                                                            ELR21790
```

```
      MMM1=MM-1                                                          ELR21800
      Y1=YR                                                              ELR21810
      YR=DABS(HR(MM,MMM1))+DABS(HI(MM,MMM1))                             ELR21820
      XI=ZZR                                                             ELR21840
      ZZR=XR                                                             ELR21850
      XR=DABS(HR(MMM1,MMM1))+DABS(HI(MMM1,MMM1))                         ELR21860
      IF (YR .LE. EPS*ZZR/YI*(ZZR+XR+XI)) GO TO 85                       ELR21880
   75 CONTINUE                                                           ELR21890
   80 MM=M                                                               ELR21900
C                                     TRIANGULAR DECOMPOSITION H=L*R     ELR21910
   85 MP1=MM+1                                                           ELR21920
      DO 110 I=MP1,NN                                                    ELR21930
      IM1=I-1                                                            ELR21940
      XR=HR(IM1,IM1)                                                     ELR21950
      XI=HI(IM1,IM1)                                                     ELR21960
      YR=HR(I,IM1)                                                       ELR21970
      YI=HI(I,IM1)                                                       ELR21980
      IF (DABS(XR)+DABS(XI) .GE. DABS(YR)+DABS(YI)) GO TO 95             ELR21990
C                                     INTERCHANGE ROWS OF HR AND HI      ELR22010
      DO 90 J=IM1,N                                                      ELR22020
      ZZR=HR(IM1,J)                                                      ELR22030
      HR(IM1,J)=HR(I,J)                                                  ELR22040
      HR(I,J)=ZZR                                                        ELR22050
      ZZI=HI(IM1,J)                                                      ELR22060
      HI(IM1,J)=HI(I,J)                                                  ELR22070
      HI(I,J)=ZZI                                                        ELR22080
   90 CONTINUE                                                           ELR22090
      Z=X/Y                                                              ELR22100
      WR(I)=ONE                                                          ELR22110
      GO TO 100                                                          ELR22120
   95 Z=Y/X                                                              ELR22130
      WR(I)=-ONE                                                         ELR22140
  100 HR(I,IM1)=ZZR                                                      ELR22150
      HI(I,IM1)=ZZI                                                      ELR22160
      DO 105 J=I,N                                                       ELR22170
      HR(I,J)=HR(I,J)-ZZR*HR(IM1,J)+ZZI*HI(IM1,J)                        ELR22180
      HI(I,J)=HI(I,J)-ZZR*HI(IM1,J)-ZZI*HR(IM1,J)                        ELR22190
  105 CONTINUE                                                           ELR22200
  110 CONTINUE                                                           ELR22210
C                                     COMPOSITION R*L=H                  ELR22220
      DO 140 J=MP1,NN                                                    ELR22230
      JM1=J-1                                                            ELR22240
      XR=HR(J,JM1)                                                       ELR22250
      XI=HI(J,JM1)                                                       ELR22260
      HR(J,JM1)=ZERO                                                     ELR22270
      HI(J,JM1)=ZERO                                                     ELR22280
C                                     INTERCHANGE COLUMNS OF HR, HI,     ELR22290
C                                       ZR, AND ZI IF NECESSARY          ELR22300
```

```
      IF (WR(J) .LE. ZERO) GO TO 125                                    ELR22310
      DO 115 I=1,J                                                      ELR22320
         ZZR=HR(I,JM1)                                                  ELR22330
         HR(I,JM1)=HR(I,J)                                              ELR22340
         HR(I,J)=ZZR                                                    ELR22350
         ZZI=HI(I,JM1)                                                  ELR22360
         HI(I,JM1)=HI(I,J)                                              ELR22370
         HI(I,J)=ZZI                                                    ELR22380
  115 CONTINUE                                                          ELR22390
      DO 120 I=K,L                                                      ELR22400
         ZZR=ZR(I,JM1)                                                  ELR22410
         ZR(I,JM1)=ZR(I,J)                                              ELR22420
         ZR(I,J)=ZZR                                                    ELR22430
         ZZI=ZI(I,JM1)                                                  ELR22440
         ZI(I,JM1)=ZI(I,J)                                              ELR22450
         ZI(I,J)=ZZI                                                    ELR22460
  120 CONTINUE                                                          ELR22470
C                                        END INTERCHANGE COLUMNS        ELR22480
  125 DO 130 I=1,J                                                      ELR22490
         HR(I,JM1)=HR(I,JM1)+XR*HR(I,J)-XI*HI(I,J)                      ELR22500
         HI(I,JM1)=HI(I,JM1)+XR*HI(I,J)+XI*HR(I,J)                      ELR22510
  130 CONTINUE                                                          ELR22520
      DO 135 I=K,L                                                      ELR22530
         ZR(I,JM1)=ZR(I,JM1)+XR*ZR(I,J)-XI*ZI(I,J)                      ELR22540
         ZI(I,JM1)=ZI(I,JM1)+XR*ZI(I,J)+XI*ZR(I,J)                      ELR22550
  135 CONTINUE                                                          ELR22560
C                                        END ACCUMULATE TRANSFORMATIONS ELR22570
  140 CONTINUE                                                          ELR22580
      GO TO 40                                                          ELR22590
C                                        A ROOT FOUND                   ELR22600
  145 WR(NN)=HR(NN,NN)+TR                                               ELR22610
      WI(NN)=HI(NN,NN)+TI                                               ELR22620
      NN=NNM1                                                           ELR22630
      GO TO 35                                                          ELR22640
C                                        ALL ROOTS FOUND. BACKSUBSTITUTE TO ELR22650
C                                          FIND VECTORS OF UPPER TRIANGULAR ELR22660
C                                          FORM                          ELR22670
  150 IF (N .EQ. 1) GO TO 9005                                          ELR22680
      FNORM=ZERO                                                        ELR22690
      DO 160 I=1,N                                                      ELR22700
         FNORM=FNORM+DABS(WR(I))+DABS(WI(I))                            ELR22710
         IF (I .EQ. N) GO TO 160                                        ELR22730
         IP1=I+1                                                        ELR22740
         DO 155 J=IP1,N                                                 ELR22750
            FNORM=FNORM+DABS(HR(I,J))+DABS(HI(I,J))                     ELR22760
  155    CONTINUE                                                       ELR22780
  160 CONTINUE                                                          ELR22790
      IF (FNORM .EQ. ZERO) GO TO 9005                                   ELR22800
```

```
C                                          DO NN=N,2,-1                                  ELR22810
      NP2 = N+2                                                                         ELR22820
      DO 180 NM=2,N                                                                     ELR22830
         NN=NP2-NM                                                                      ELR22840
         XR=WR(NN)                                                                      ELR22850
         XI=WI(NN)                                                                      ELR22860
         NNM1=NN-1                                                                      ELR22870
C                                          DO I=NN-1,1,-1                                ELR22880
         DO 175 II=1,NNM1                                                               ELR22890
            I=NN-II                                                                     ELR22900
            ZZR=HR(I,NN)                                                                ELR22910
            ZZI=HI(I,NN)                                                                ELR22920
            IF (I .EQ. NNM1) GO TO 170                                                  ELR22930
            IP1=I+1                                                                     ELR22940
            DO 165 J=IP1,NNM1                                                           ELR22950
               ZZR=ZZR+HR(I,J)*HR(J,NN)-HI(I,J)*HI(J,NN)                                ELR22960
               ZZI=ZZI+HR(I,J)*HI(J,NN)+HI(I,J)*HR(J,NN)                                ELR22970
  165       CONTINUE                                                                    ELR22980
  170       YR=XR-WR(I)                                                                 ELR22990
            YI=XI-WI(I)                                                                 ELR23000
            IF (YR .EQ. ZERO .AND. YI .EQ. ZERO) YR=EPS*FNORM                           ELR23010
            Z=Z/Y                                                                       ELR23020
            HR(I,NN)=T3(1)                                                              ELR23030
            HI(I,NN)=T3(2)                                                              ELR23040
  175    CONTINUE                                                                       ELR23050
  180 CONTINUE                                                                          ELR23060
C                                          END BACKSUBSTITUTION                          ELR23070
      NM1=N-1                                                                           ELR23080
C                                          VECTOR OF ISOLATED ROOTS                      ELR23090
      DO 190 I=1,NM1                                                                    ELR23100
         IF (I .GE. K .AND. I .LE. L) GO TO 190                                         ELR23110
         IP1=I+1                                                                        ELR23120
         DO 185 J=IP1,N                                                                 ELR23130
            ZR(I,J)=HR(I,J)                                                             ELR23140
            ZI(I,J)=HI(I,J)                                                             ELR23150
  185    CONTINUE                                                                       ELR23160
  190 CONTINUE                                                                          ELR23170
      IF (L .EQ. 0) GO TO 9005                                                          ELR23180
C                                          MULTIPLY BY TRANSFORMATION MATRIX             ELR23190
C                                            TO GIVE VECTORS OF ORIGINAL FULL            ELR23200
C                                            MATRIX                                      ELR23210
      NPL=N+K                                                                           ELR23220
C                                          DO J=N,K+1,-1                                 ELR23230
      DO 200 JJ=K,NM1                                                                   ELR23240
         J=NPL-JJ                                                                       ELR23250
         JM1=J-1                                                                        ELR23260
         DO 200 I=K,L                                                                   ELR23270
            ZZR=ZR(I,J)                                                                 ELR23280
```

```
                ZZI=ZI(I,J)                                                     ELR23290
                MM=JM1                                                          ELR23300
                IF (L .LT. J) MM=L                                              ELR23310
                DO 195 M=K,MM                                                   ELR23320
                   ZZR=ZZR+ZR(I,M)*HR(M,J)-ZI(I,M)*HI(M,J)                      ELR23330
                   ZZI=ZZI+ZR(I,M)*HI(M,J)+ZI(I,M)*HR(M,J)                      ELR23340
  195           CONTINUE                                                        ELR23350
                ZR(I,J)=ZZR                                                     ELR23360
                ZI(I,J)=ZZI                                                     ELR23370
  200 CONTINUE                                                                  ELR23380
      GO TO 9005                                                                ELR23390
C                                       SET ERROR - NO CONVERGENCE TO AN        ELR23400
C                                         EIGENVALUE AFTER 30 ITERATIONS        ELR23410
  205 IER=129                                                                   ELR23420
      INFER=NN                                                                  ELR23430
 9000 CONTINUE                                                                  ELR23440
      CALL UERTST (IER,6HELRH2C)                                                ELR23450
 9005 RETURN                                                                    ELR23460
      END                                                                       ELR23470
C
CCC
CCC     SUBROUTINE EBBCKC (ZR,ZI,N,IZ,K,L,M,D)                                  EBBC0010
CCC                                                                             EBBC0020
C-EBBCKC--------D-------LIBRARY 1-------------------------------------------    EBBC0030
CCC                                                                             EBBC0040
CCC   FUNCTION              - BACKTRANSFORM THE EIGENVECTORS OF A BALANCED      EBBC0050
CCC                           COMPLEX GENERAL MATRIX.                           EBBC0060
CCC   USAGE                 - CALL EBBCKC (ZR,ZI,N,IZ,K,L,M,D)                  EBBC0070
CCC   PARAMETERS    ZR      - INPUT/OUTPUT MATRICES OF DIMENSION N BY M.        EBBC0080
CCC                 ZI        ON INPUT, THE FIRST M COLUMNS OF ZR AND           EBBC0090
CCC                           ZI CONTAIN THE REAL AND IMAGINARY PARTS,          EBBC0100
CCC                           RESPECTIVELY, OF THE EIGENVECTORS TO BE           EBBC0110
CCC                           BACK TRANSFORMED. ON OUTPUT, THESE M              EBBC0120
CCC                           COLUMNS CONTAIN THE REAL AND IMAGINARY            EBBC0130
CCC                           PARTS OF THE TRANSFORMED EIGENVECTORS.            EBBC0140
CCC                 N       - INPUT SCALAR CONTAINING THE NUMBER OF           EBBC0150
CCC                           ROWS IN THE MATRIX Z = (ZR,ZI). N MUST            EBBC0160
CCC                           NOT BE GREATER THAN IZ.                           EBBC0170
CCC                 IZ      - INPUT SCALAR CONTAINING THE ROW DIMENSION        EBBC0180
CCC                           OF MATRICES ZR AND ZI IN THE CALLING              EBBC0190
CCC                           PROGRAM.                                          EBBC0200
CCC                 K       - INPUT SCALARS CONTAINING THE BOUNDARY           EBBC0210
CCC                 L         INDICES FOR THE BALANCED MATRIX. K AND L        EBBC0220
CCC                           ARE TWO OUTPUT PARAMETERS FROM IMSL ROUTINE       EBBC0230
CCC                           EBALAC.                                           EBBC0240
CCC                 M       - INPUT SCALAR CONTAINING THE NUMBER OF COLUMNS     EBBC0250
CCC                           OF Z = (ZR,ZI) TO BE BACK TRANSFORMED.            EBBC0260
C                   D       - INPUT VECTOR OF LENGTH N CONTAINING THE         EBBC0270
```

```
C                                  DETAILS OF THE TRANSFORMATIONS PRODUCED      EBBC0280
CC                                 BY IMSL ROUTINE EBALAC.                      EBBC0290
CC   PRECISION              - SINGLE/DOUBLE                                     EBBC0300
CC   LANGUAGE               - FORTRAN                                           EBBC0310
CC---------------------------------------------------------------------------EBBC0320
CC   LATEST REVISION        - MARCH 9, 1977                                     EBBC0330
C                                                                              EBBC0340
      SUBROUTINE EBBCKC (ZR,ZI,N,IZ,K,L,M,D)                                   EBBC0350
C                                                                              EBBC0360
      DIMENSION           ZR(IZ,1),ZI(IZ,1),D(1)                               EBBC0370
      DOUBLE PRECISION    ZR,ZI,D,S                                            EBBC0380
      IF (L .EQ. K) GO TO 15                                                   EBBC0390
      DO 10 I = K,L                                                            EBBC0400
         S = D(I)                                                              EBBC0410
C                                        LEFT HAND EIGENVECTORS ARE BACK       EBBC0420
CC                                         TRANSFORMED IF THE ABOVE            EBBC0430
C                                          STATEMENT IS REPLACED BY S=1.0/D(I) EBBC0440
         DO 5 J = 1,M                                                          EBBC0450
            ZR(I,J) = ZR(I,J)*S                                                EBBC0460
            ZI(I,J) = ZI(I,J)*S                                                EBBC0470
    5    CONTINUE                                                              EBBC0480
   10 CONTINUE                                                                 EBBC0490
C                                        DO 25 I=K-1,1,-1 AND                  EBBC0500
CC                                         DO I=L+1,N,1                        EBBC0510
   15 DO 25 II = 1,N                                                           EBBC0520
         I = II                                                                EBBC0530
         IF (I .GE. K .AND. I .LE. L) GO TO 25                                 EBBC0540
         IF (I .LT. K) I = K-II                                                EBBC0550
         KK = D(I)                                                             EBBC0560
         IF (KK .EQ. I) GO TO 25                                               EBBC0570
         DO 20 J = 1,M                                                         EBBC0580
            S = ZR(I,J)                                                        EBBC0590
            ZR(I,J) = ZR(KK,J)                                                 EBBC0600
            ZR(KK,J) = S                                                       EBBC0610
            S = ZI(I,J)                                                        EBBC0620
            ZI(I,J) = ZI(KK,J)                                                 EBBC0630
            ZI(KK,J) = S                                                       EBBC0640
   20    CONTINUE                                                              EBBC0650
   25 CONTINUE                                                                 EBBC0660
      RETURN                                                                   EBBC0670
      END                                                                      EBBC0680
C                                                                              
CCC                                                                            
      SUBROUTINE UERTST (IER,NAME)                                             UERT0010
CC                                                                             UERT0020
C-UERTST----------------LIBRARY 1-----------------------------------------UERT0030
CC                                                                             UERT0040
C   FUNCTION                - ERROR MESSAGE GENERATION                         UERT0050
```

```
C    USAGE                - CALL UERTST(IER,NAME)                                    UERT0060
C    PARAMETERS   IER     - ERROR PARAMETER. TYPE + N  WHERE                         UERT0070
C                             TYPE= 128 IMPLIES TERMINAL ERROR                       UERT0080
C                                   64 IMPLIES WARNING WITH FIX                      UERT0090
C                                   32 IMPLIES WARNING                               UERT0100
C                               N   = ERROR CODE RELEVANT TO CALLING ROUTINE         UERT0110
C                    NAME    - INPUT VECTOR CONTAINING THE NAME OF THE              UERT0120
C                              CALLING ROUTINE AS A SIX CHARACTER LITERAL            UERT0130
C                              STRING.                                               UERT0140
C    LANGUAGE             - FORTRAN                                                  UERT0150
C-------------------------------------------------------------------------------    UERT0160
C    LATEST REVISION      - JANUARY 18, 1974                                         UERT0170
C                                                                                    UERT0180
      SUBROUTINE UERTST(IER,NAME)                                                    UERT0190
C                                                                                    UERT0200
      DIMENSION            ITYP(5,4),IBIT(4)                                         UERT0210
      INTEGER*2            NAME(3)                                                   UERT0220
      INTEGER              WARN,WARF,TERM,PRINTR                                     UERT0230
      EQUIVALENCE          (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)              UERT0240
      DATA      ITYP       /'WARN','ING ','    ','    ','    ',                     UERT0250
     *                      'WARN','ING(','WITH',' FIX',')   ',                     UERT0260
     *                      'TERM','INAL','    ','    ','    ',                     UERT0270
     *                      'NON-','DEFI','NED ','    ','    '/,                    UERT0280
     *           IBIT       / 32,64,128,0/                                          UERT0290
      DATA      PRINTR     / 6/                                                      UERT0300
      IER2=IER                                                                       UERT0310
      IF (IER2 .GE. WARN) GO TO 5                                                    UERT0320
C                                      NON-DEFINED                                   UERT0330
      IER1=4                                                                         UERT0340
      GO TO 20                                                                       UERT0350
    5 IF (IER2 .LT. TERM) GO TO 10                                                   UERT0360
C                                      TERMINAL                                      UERT0370
      IER1=3                                                                         UERT0380
      GO TO 20                                                                       UERT0390
   10 IF (IER2 .LT. WARF) GO TO 15                                                   UERT0400
C                                      WARNING(WITH FIX)                             UERT0410
      IER1=2                                                                         UERT0420
      GO TO 20                                                                       UERT0430
C                                      WARNING                                       UERT0440
   15 IER1=1                                                                         UERT0450
C                                      EXTRACT 'N'                                   UERT0460
   20 IER2=IER2-IBIT(IER1)                                                           UERT0470
C                                      PRINT ERROR MESSAGE                           UERT0480
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,5),NAME,IER2,IER                           UERT0490
   25 FORMAT(' *** I M S L(UERTST) ***  ',5A4,4X,3A2,4X,I2,                         UERT0500
     *    ' (IER = ',I3,')')                                                        UERT0510
      RETURN                                                                         UERT0520
      END                                                                            UERT0530
```

```
//EIG$FCN JOB (1719,0947,AX74),'SMC 1882',TIME=2
// EXEC FORTCLGW
//FORT.SYSIN DD *
C.....EIGF  10
C                                                                       EIGF  20
C     PROGRAM EIGFCN                                                    EIGF  30
C                                                                       EIGF  40
C     PERTURBATION VELOCITY PLOT PROGRAM                                EIGF  50
C           NI = 0                                                      EIGF  60
C                                                                       EIGF  70
C                                                                       EIGF  80
C     PURPOSE                                                           EIGF  90
C                                                                       EIGF 100
C        TO PLOT THE NONDIMENSIONALIZED PERTURBATION VELOCITY U AGAINST EIGF 110
C             NONDIMENSIONALIZED RADIUS UTILIZING THE DATA GENERATED    EIGF 120
C             BY PROGRAM PIPE0 (MODE=1).  PLOTTING IS PERFORMED ON      EIGF 130
C             THE NPS VERSATEC PLOTTER USING SUBROUTINE PLOTG.          EIGF 140
C                                                                       EIGF 150
C                                                                       EIGF 160
C.....EIGF 170
C                                                                       EIGF 180
C                                                                       EIGF 190
      IMPLICIT REAL*8(A-H,O-Z)                                          EIGF 200
      COMPLEX *16QPRIM(85),DQPRIM(85),ALPHA,UP(85),UPPRIM(85),CONST,GAMMEIGF 210
     1A                                                                 EIGF 220
      REAL *8ETA(85),U(85),UR(85),UI(85)                                EIGF 230
      REAL *4UR1(85),UI1(85),RAD1(85),SREY,SLAMDA,SGAMMA,AR,AI          EIGF 240
C                                                                       EIGF 250
C     READ N,REY,ALPHA,LAMBDA,GAMMA & QPRIME'S                          EIGF 260
C                                                                       EIGF 270
      READ (5,6) N,REY,ALPHA                                            EIGF 280
      NO = N+1                                                          EIGF 290
      N1 = N+2                                                          EIGF 300
      READ (5,7) AMDA,GAMMA                                             EIGF 310
      READ (5,8) KSET                                                   EIGF 320
      SLAMDA = AMDA                                                     EIGF 340
      SREY = REY                                                        EIGF 330
      AR = ALPHA                                                        EIGF 350
      AI = AIMAG(ALPHA)                                                 EIGF 360
      SGAMMA = GAMMA                                                    EIGF 370
      SGAMMA = SGAMMA+AR                                                EIGF 380
C                                                                       EIGF 390
C                                                                       EIGF 400
      DO 1 I=2,NO                                                       EIGF 410
      READ (5,9) ETA(I),QPRIM(I)                                        EIGF 420
    1 CONTINUE                                                          EIGF 430
C                                                                       EIGF 440
C                                                                       EIGF 450
```

```
C       COMPUTE UPPRIME'S                                                       EIGF  460
C                                                                               EIGF  470
        DEL = 1DO/DFLOAT(N+1)                                                   EIGF  480
        ETA(1) = 0.0DO                                                          EIGF  490
        QPRIM(1) = 1.795918367DO*QPRIM(2)-1.24781341DO*QPRIM(3)+0.60641399EIGF  500
       14DO*QPRIM(4)-0.177842566DO*QPRIM(5)+0.023323615DO*QPRIM(6)              EIGF  510
        UPPRIM(1) = 2DO*QPRIM(1)                                                FIGF  520
C                                                                               EIGF  530
        CALL COEFNT (ETA(2),AMDA,COEF,KSET)                                     EIGF  540
        DQPRIM(2) = 2DO*(-QPRIM(2)+QPRIM(3))/(3DO*DEL)                          EIGF  550
        DQPRIM(2) = COEF*DQPRIM(2)                                              EIGF  560
        UPPRIM(2) = 2DO*QPRIM(2)+ETA(2)*DQPRIM(2)                               EIGF  570
C                                                                               EIGF  580
        CALL COEFNT (ETA(3),AMDA,COEF,KSET)                                     EIGF  590
        DQPRIM(3) = 4DO*(-QPRIM(2)+QPRIM(3))/(3DO*DEL)                          EIGF  600
        DQPRIM(3) = COEF*DQPRIM(3)                                              EIGF  610
        UPPRIM(3) = 2DO*QPRIM(3)+ETA(3)*DQPRIM(3)                               EIGF  620
C                                                                               EIGF  630
C                                                                               EIGF  640
        DO 2 I=4,N                                                              EIGF  650
        CALL COEFNT (ETA(I),AMDA,COEF,KSET)                                     EIGF  660
        DQPRIM(I) = (QPRIM(I+1)-QPRIM(I-1))/(2DO*DEL)                           EIGF  670
        DQPRIM(I) = COEF*DQPRIM(I)                                              EIGF  680
        UPPRIM(I) = 2DO*QPRIM(I)+ETA(I)*DQPRIM(I)                               EIGF  690
      2 CONTINUE                                                                EIGF  700
C                                                                               EIGF  710
C                                                                               EIGF  720
        CALL COEFNT (ETA(NO),AMDA,COEF,KSET)                                    EIGF  730
        DQPRIM(NO) = -QPRIM(N)/(2DO*DEL)                                        EIGF  740
        DQPRIM(NO) = COEF*DQPRIM(NO)                                            EIGF  750
        UPPRIM(NO) = 2DO*QPRIM(NO)+ETA(NO)*DQPRIM(NO)                           EIGF  760
C                                                                               EIGF  770
        ETA(N1) = 1.0DO                                                         EIGF  780
        UPPRIM(N1) = (0DO,0DO)                                                  EIGF  790
C                                                                               EIGF  800
        WRITE (6,10)                                                            EIGF  810
C                                                                               EIGF  820
C       DETERMINE U VECTOR OF LARGEST MAGNITUDE                                 EIGF  830
C                                                                               EIGF  840
        C = 0.DO                                                                EIGF  850
C                                                                               EIGF  860
        DO 3 I=1,N1                                                             EIGF  870
        IF (CDABS(UPPRIM(I)).GT.C) INDEX=I                                      EIGF  880
        IF (CDABS(UPPRIM(I)).GT.C) C=CDABS(UPPRIM(I))                           EIGF  890
      3 CONTINUE                                                                EIGF  900
C                                                                               EIGF  910
C                                                                               EIGF  920
        CONST = DCONJG(UPPRIM(INDEX))/C**2                                      EIGF  530
```

```
C                                                                       EIGF 940
CC      NORMALIZE UPPRIMES'S AND SPLIT INTO REAL & IMAGINARY VECTORS    EIGF 950
CC                                                                      EIGF 960
C                                                                       EIGF 970
        DO 4 I=1,N1                                                     EIGF 980
        UP(I) = CONST*UPPRIM(I)                                         EIGF 990
        UR(I) = UP(I)                                                   EIGF1000
        UI(I) = (0D0,-1D0)*UP(I)                                        EIGF1010
      4 CONTINUE                                                        EIGF1020
C                                                                       EIGF1030
CC                                                                      EIGF1040
CC      CONVERT U'S AND ETA'S TO SINGLE PRECISION FOR PLOTG             EIGF1050
CC                                                                      EIGF1060
C                                                                       EIGF1070
        DO 5 I=1,N1                                                     EIGF1080
        RAD1(I) = ETA(I)                                                EIGF1090
        UR1(I) = UR(I)                                                  EIGF1100
        UI1(I) = UI(I)                                                  EIGF1110
        WRITE (6,11) RAD1(I),UR1(I),UI1(I)                              EIGF1120
      5 CONTINUE                                                        EIGF1130
C                                                                       EIGF1140
CCC                                                                     EIGF1150
CCC     PLOT RESULTS                                                    EIGF1160
C                                                                       EIGF1170
        CALL PLOTG(RAD1,UR1,N1,1,1,1,'RADIUS',6,'PERTURBATION VELOCITY',EIGF1180
     $ 21,0.,1.,-1.,1.,7.,7.)                                           EIGF1190
        CALL PLOTG(RAD1,UI1,N1,2,1,5,'RADIUS',6,'PERTURBATION VELOCITY',EIGF1200
     $ 21,0.,1.,-1.,1.,7.,7.)                                           EIGF1210
        CALL CHART (N,SREY,AR,AI,SGAMMA,SLAMDA)                         EIGF1220
        CALL PLOT (0.0,0.0,999)                                         EIGF1230
        STOP                                                            EIGF1240
C                                                                       EIGF1250
      6 FORMAT (I2,3D20.10)                                             EIGF1260
      7 FORMAT (F15.7,2(1PD20.10))                                      EIGF1270
      8 FORMAT (I2)                                                     EIGF1280
      9 FORMAT (F15.7,2(1PD20.10))                                      EIGF1290
     10 FORMAT ('1')                                                    EIGF1300
     11 FORMAT (' ',3F15.7)                                             EIGF1310
        END                                                             EIGF1320
C
CCC.................SUBROUTINE COEFNT(ETA,AMDA,COEF,KSET).............COEF  10
CCC                                                                     COEF  20
CCC                                                                     COEF  30
CCC     PURPOSE--WHEN AN OFFSET MESH IS USED, THIS SUBROUTINE GENERATES COEF  40
CCC             THE COEFFICIENT REQUIRED TO CONVERT DQ/DETA TO DQ/DR AND COEF  50
CCC             CONVERTS THE UNIFORM ETA VALUE INTO THE NONUNIFORM R VALUECOEF  60
C                                                                       COEF  70
```

```
C                                                                       COEF  80
CC     EXAMPLE OF THE CALLING ARGUMENT                                  COEF  90
CCC                                                                     COEF 100
CCC             CALL COEFNT(ETA,AMDA,COEF,KSET)                         COEF 110
CCC                                                                     COEF 120
CCC                                                                     COEF 130
CC     DESCRIPTION OF PARAMETERS                                        COEF 140
CC                                                                      COEF 150
CC        ETA -THE VALUE OF THE INDEPENDENT VARIABLE REPLACING R        COEF 160
CC             IN THE NONUNIFORM MESH FOR THE STATION BEING COMPUTED.   COEF 170
CC                                                                      COEF 180
CC        AMDA-THE NONDIMENSIONAL MESH PARAMETER FOR THE PARTICULAR     COEF 190
CC             SET OF DATA BEING PLOTTED.                               COEF 200
CC                                                                      COEF 210
CC        COEF-THE VALUE OF (DR/DETA)**-1 NEEDED TC CONVERT DQ/DETA TO  COEF 220
CC             DQ/DR.                                                   COEF 230
CC                                                                      COEF 240
CC        KSET-OFFSET PARAMETER WHICH IS EQUAL TO -1 FOR WALL OFFSET,   COEF 250
CC             0 FOR UNIFORM MESH AND 1 FOR AXIS OFFSET.                COEF 260
CC                                                                      COEF 270
CC                                                                      COEF 280
CC     OTHER SUBROUTINES NEEDED                                         COEF 290
CC                                                                      COEF 300
CC        NONE                                                          COEF 310
CC                                                                      COEF 320
C*************************************************************************COEF 330
C                                                                       COEF 340
C                                                                       COEF 350
      SUBROUTINE COEFNT (ETA,AMDA,COEF,KSET)                            COEF 360
C                                                                       COEF 370
CC                                                                      COEF 380
      IMPLICIT REAL*8(A-H,O-Z)                                          COEF 390
C                                                                       COEF 400
      TETA = ETA                                                        COEF 410
      IF (AMDA.LE.1D-10) GO TO 3                                        COEF 420
      IF (KSET.EQ.1) TETA=1D0-TETA                                      COEF 430
      CNST = DTANH(AMDA)/AMDA                                           COEF 440
      ETAP = AMDA*TETA                                                  COEF 450
      COEF = CNST*(DCOSH(ETAP))**2                                      COEF 460
      IF (AMDA.GE.1.D-10) CONST=1D0/DTANH(AMDA)                         COEF 470
      IF (KSET) 1,3,2                                                   COEF 480
C                                                                       COEF 490
    1 ETA = CONST*DTANH(ETAP)                                           COEF 500
      RETURN                                                            COEF 510
C                                                                       COEF 520
    2 ETA = 1.D0-CONST*DTANH(ETAP)                                      COEF 530
      RETURN                                                            COEF 540
C                                                                       COEF 550
```

```
    3 COEF = 1D0                                                          COEF  560
      RETURN                                                              COEF  570
      END                                                                 COEF  580
C
CC
CC.................SUBROUTINE CHART(N,SREY,AR,AI,SGAMMA,SLAMDA).............CHAR   10
CC                                                                        CHAR   20
CC     PURPOSE                                                            CHAR   30
CC                                                                        CHAR   40
CC        TO LABEL THE GRAPH WITH INFORMATION PERTAINING TO THE PLOT       CHAR   50
CC                                                                        CHAR   60
CC                                                                        CHAR   70
CC     EXAMPLE OF THE CALLING ARGUMENT                                    CHAR   80
CC                                                                        CHAR   90
CC        CALL CHART(N,SREY,AR,AI,SGAMMA,SLAMDA)                          CHAR  100
CC                                                                        CHAR  110
CC                                                                        CHAR  120
CC     DESCRIPTION OF PARAMETERS                                          CHAR  130
CC                                                                        CHAR  140
CC        THE PARAMETERS ARE SELF-EXPLANATORY AND MUST BE IN SINGLE       CHAR  150
CC             PRECISION FOR PLOTTING.                                    CHAR  160
CC                                                                        CHAR  170
CC                                                                        CHAR  180
CC     OTHER SUBROUTINES NEEDED                                           CHAR  190
CC                                                                        CHAR  200
CC        ONLY BUILT-IN VERSATEC PLOTTING FUNCTIONS NEWPEN,SYMBOL &       CHAR  210
CC             NUMBER.  NOTE THAT THESE ROUTINES MAY ONLY BE              CHAR  220
CC             ACCESSED WHEN RUNNING UNDER 'FORTCLGW'.                    CHAR  230
CC                                                                        CHAR  240
CC                                                                        CHAR  250
CC.........................................................................CHAR  260
CC                                                                        CHAR  270
C                                                                         CHAR  280
      SUBROUTINE CHART (N,SREY,AR,AI,SGAMMA,SLAMDA)                       CHAR  290
C                                                                         CHAR  300
CC                                                                        CHAR  310
      X0 = 2.5                                                            CHAR  320
      Y0 = 6.5                                                            CHAR  330
      HT = 0.15                                                           CHAR  340
      HT1 = 0.7*HT                                                        CHAR  350
      DELY1 = .08+HT                                                      CHAR  360
      DELY2 = .065+HT1                                                    CHAR  370
      DELX = .1                                                           CHAR  380
C                                                                         CHAR  390
CC    GRAPH TITLE                                                         CHAR  400
C                                                                         CHAR  410
      CALL NEWPEN (2)                                                     CHAR  420
      CALL SYMBOL(X0,Y0,HT,'NORMALIZED PERTURBATION VELOCITY',0.,32)      CHAR  430
```

```
      XO = XO+7.*DELX                                                    CHAR 440
      YO = YO-DELY1                                                      CHAR 450
      CALL SYMBOL(XO,YO,HT,'FOR THE CASE N = 0',0.,18)                   CHAR 460
C                                                                        CHAR 470
CC    MESH VALUE                                                         CHAR 480
C                                                                        CHAR 490
      CALL NEWPEN (1)                                                    CHAR 500
      XO = XO+4.*DELX                                                    CHAR 510
      YO = YO-DELY1                                                      CHAR 520
      SN = FLOAT(N)                                                      CHAR 530
      CALL SYMBOL(XO,YO,HT1,'NMESH  = ',0.,9)                            CHAR 540
      CALL NUMBER (999.,999.,HT1,SN,0.,-1)                               CHAR 550
C                                                                        CHAR 560
CC    REY VALUE                                                          CHAR 570
C                                                                        CHAR 580
      YC = YO-DELY2                                                      CHAR 590
      CALL SYMBOL(XO,YO,HT1,'REY    = ',0.,9)                            CHAR 600
      CALL NUMBER (999.,999.,HT1,SREY,0.,-1)                             CHAR 610
C                                                                        CHAR 620
CC    ALPHA VALUE                                                        CHAR 630
C                                                                        CHAR 640
      X1 = XO+11.*DELY2                                                  CHAR 650
      YO = YO-DELY2                                                      CHAR 660
      CALL SYMBOL(XO,YO,HT1,'ALPHA  = ',0.,9)                            CHAR 670
      CALL NUMBER (999.,999.,HT1,AR,0.,1)                                CHAR 680
      CALL NUMBER (X1,YO,HT1,AI,0.,1)                                    CHAR 690
C                                                                        CHAR 700
CC    GAMMA RL* VALUE                                                    CHAR 710
C                                                                        CHAR 720
      YO = YO-DELY2                                                      CHAR 730
      CALL SYMBOL(XO,YO,HT1,'GAMMA* = ',0.,9)                            CHAR 740
      CALL NUMBER (999.,999.,HT1,SGAMMA,0.,4)                            CHAR 750
C                                                                        CHAR 760
CC    LAMBDA VALUE                                                       CHAR 770
C                                                                        CHAR 780
      YO = YO-DELY2                                                      CHAR 790
      CALL SYMBOL(XO,YO,HT1,'LAMBDA = ',0.,9)                            CHAR 800
      CALL NUMBER (999.,999.,HT1,SLAMDA,0.,1)                            CHAR 810
C                                                                        CHAR 820
CC    SYMBOL LEGEND                                                      CHAR 830
C                                                                        CHAR 840
      YO = 1.75                                                          CHAR 850
      CALL SYMBOL(XO,YO,HT1,'OCTAGON = U(REAL)',0.,17)                   CHAR 860
      YO = YO-DELY2                                                      CHAR 870
      CALL SYMBOL(XO,YO,HT1,'DIAMOND = U(IMAG)',0.,17)                   CHAR 880
      RETURN                                                             CHAR 890
      END                                                                CHAR 900
C
```

```
C.....
C
C     THE FOLLOWING CARDS COMPRISE THE DATA DECK FOR PROGRAM EIGFCN.
C        /*
C        //GO.SYSIN DD *
C                 .
C                 .
C        DATA DECK FROM ONE RUN OF PROGRAM PIPEO (MODENO = 1)
C                 .
C                 .
C        /*
C
C.....
```

```
//STB$CONT JOB (1719,0947,AX74),'SMC 1882',TIME=2
// EXEC FORTCLGW
//FORT.SYSIN DD *
C.....STBC  10
C                                                                      STBC  20
C     PROGRAM STBCONT                                                  STBC  30
C                                                                      STBC  40
C     PURPOSE                                                          STBC  50
C                                                                      STBC  60
C        TO GENERATE A CONTOUR PLOT CONTAINING LINES OF INCIPIENT,     STBC  70
C        CRITICAL & FULLY DEVELOPED INSTABILITY USING DATA GENERATED   STBC  80
C        BY PROGRAM PIPEO (MODENO = 2).  THE PLOT IS GENERATED ON THE  STBC  90
C        NPS VERSATEC PLOTTER USING SUBROUTINE PLOTG WITH ALPHA REAL   STBC 100
C        ON THE X-AXIS AND ALPHA IMAGINARY ON THE Y-AXIS.              STBC 110
C                                                                      STBC 120
C.....STBC 130
C                                                                      STBC 140
      DIMENSION X1(200), X2(200), X3(200), Y1(200), Y2(200), Y3(200)   STBC 150
      COMMON /ARAY/ G1(41,41),AR1(41),AI1(41)                          STBC 160
      DATA X1,X2,X3,Y1,Y2,Y3/1200*0.0/                                 STBC 170
C                                                                      STBC 180
CC    SET INITIAL VALUES                                               STBC 190
C                                                                      STBC 200
      READ (5,9) NDIM                                                  STBC 210
      XMIN = -.5                                                       STBC 220
      XMAX = 0.                                                        STBC 230
      YMIN = 0.                                                        STBC 240
      YMAX = 10.                                                       STBC 250
C                                                                      STBC 260
CCC   THE NEXT 3 VALUES MUST BE SET BY THE USER PRIOR TO RUNNING       STBC 270
CCC   THE PROGRAM.                                                     STBC 280
C                                                                      STBC 290
      SN = 47.                                                         STBC 300
      SREY = 4000.                                                     STBC 310
      SLAMDA = 0.0                                                     STBC 320
C                                                                      STBC 330
CCC   READ STABILITY MAP VALUES OUTPUT BY PROGRAM PIPEO (MODENO =2)    STBC 340
CC                                                                     STBC 350
C                                                                      STBC 360
      DO 1 I=1,NDIM                                                    STBC 370
C                                                                      STBC 380
      DO 1 J=1,NDIM                                                    STBC 390
      READ (5,10) AR1(I),AI1(J),G1(I,J)                                STBC 400
    1 CONTINUE                                                         STBC 410
C                                                                      STBC 420
CC                                                                     STBC 430
CCC   COMPUTE POINTS FOR INCIPIENT,CRITICAL & FULLY DEVELOPED          STBC 440
C     INSTABILITY CURVES.                                              STBC 450
```

```
C                                                                       STBC  460
      CALL SEARCH (-1,X1,Y1,NPLT1,NDIM)                                 STBC  470
      CALL SEARCH (0,X2,Y2,NPLT2,NDIM)                                  STBC  480
      CALL SEARCH (1,X3,Y3,NPLT3,NDIM)                                  STBC  490
C                                                                       STBC  500
CC    JUMP TO PLOT LABEL ROUTINE IF NO INCIPIENT POINTS                 STBC  510
C                                                                       STBC  520
      IF (NPLT1) 8,8,2                                                  STBC  530
C                                                                       STBC  540
CC    IF POINTS COMPUTED, NEW PAGE AND WRITE THEM OUT                   STBC  550
C                                                                       STBC  560
    2 WRITE (6,11)                                                      STBC  570
C                                                                       STBC  580
      DO 3 I=1,NPLT1                                                    STBC  590
      WRITE (6,12) X1(I),Y1(I)                                          STBC  600
    3 CONTINUE                                                          STBC  610
C                                                                       STBC  620
CC                                                                      STBC  630
CC    PLOT INCIPIENT INSTABILITY POINTS                                 STBC  640
C                                                                       STBC  650
      CALL PLOTG(X1,Y1,NPLT1,1,0,1,'ALPHA REAL',10,'ALPHA IMAGINARY',15,STBC  660
     $ XMIN,XMAX,YMIN,YMAX,7.,7.)                                       STBC  670
C                                                                       STBC  680
CC    LEGEND FOR INCIPIENT SYMBOL                                       STBC  690
C                                                                       STBC  700
      CALL NEWPEN (1)                                                   STBC  710
      CALL SYMBOL(1.3,0.7,.1,'OCTAGON  = INCIPIENT INSTABILITY',0.,32)  STBC  720
C                                                                       STBC  730
CC    JUMP TO PLOT LABEL ROUTINE IF NO CRITICAL POINTS                  STBC  740
C                                                                       STBC  750
      IF (NPLT2) 8,8,4                                                  STBC  760
C                                                                       STBC  770
CC    IF  POINTS COMPUTED, NEW PAGE AND PRINT THEM OUT                  STBC  780
C                                                                       STBC  790
    4 WRITE (6,11)                                                      STBC  800
C                                                                       STBC  810
      DO 5 I=1,NPLT2                                                    STBC  820
      WRITE (6,12) X2(I),Y2(I)                                          STBC  830
    5 CONTINUE                                                          STBC  840
C                                                                       STBC  850
CC                                                                      STBC  860
CC    PLOT CRITICAL POINTS                                              STBC  870
C                                                                       STBC  880
      CALL PLOTG(X2,Y2,NPLT2,2,0,2,'ALPHA REAL',10,'ALPHA IMAGINARY',15,STBC  890
     $ XMIN,XMAX,YMIN,YMAX,7.,7.)                                       STBC  900
C                                                                       STBC  910
CC    LEGEND FOR CRITICAL SYMBOL                                        STBC  920
C                                                                       STBC  930
```

```
      CALL NEWPEN (1)                                                    STBC  940
      CALL SYMBOL(1.3,.53,.1,'TRIANGLE = CRITICAL INSTABILITY',0.,31)    STBC  950
C                                                                        STBC  960
CC    JUMP TO PLOT LABEL ROUTINE IF NO FULLY DEVELOPED POINTS            STBC  970
C                                                                        STBC  980
      IF (NPLT3) 8,8,6                                                   STBC  990
C                                                                        STBC1000
CC    IF  POINTS COMPUTED, NEW PAGE AND PRINT THEM OUT                   STBC1010
C                                                                        STBC1020
    6 WRITE (6,11)                                                       STBC1030
C                                                                        STBC1040
      DO 7 I=1,NPLT3                                                     STBC1050
      WRITE (6,12) X3(I),Y3(I)                                           STBC1060
    7 CONTINUE                                                           STBC1070
C                                                                        STBC1080
CC                                                                       STBC1090
CC    PLOT FULLY DEVELOPED POINTS                                        STBC1100
C                                                                        STBC1110
      CALL PLOTG(X3,Y3,NPLT3,3,0.5,'ALPHA REAL',10,'ALPHA IMAGINARY',15, STBC1120
     $ XMIN,XMAX,YMIN,YMAX,7.,7.)                                        STBC1130
C                                                                        STBC1140
CC    LEGEND FOR FULLY DEVELOPED SYMBOL                                  STBC1150
C                                                                        STBC1160
      CALL NEWPEN (1)                                                    STBC1170
      CALL SYMBOL(1.3,.36,.1,'DIAMOND  = FULLY DEVELOPED INSTABILITY',   STBC1180
     $ C.,38)                                                            STBC1190
C                                                                        STBC1200
CC    LABEL THE PLOT                                                     STBC1210
C                                                                        STBC1220
    8 CALL CHART (SN,SREY,SLAMDA)                                        STBC1230
      CALL PLOT (0.,0.,999)                                              STBC1240
      STOP                                                               STBC1250
C                                                                        STBC1260
    9 FORMAT (I2)                                                        STBC1270
   10 FORMAT (3E20.10)                                                   STBC1280
   11 FORMAT ('1')                                                       STBC1290
   12 FORMAT (' ',2E20.10)                                               STBC1300
      END                                                                STBC1310
C
CC
CC.....SUBROUTINE SEARCH(NCASE,X,Y,NDIM).....SEAR  10
CC                                                                       SEAR  20
CC    PURPOSE                                                            SEAR  30
CC                                                                       SEAR  40
CC       TO SCAN THE STABILITY MAP FOR CHANGES OF SIGN WITH RESPECT      SEAR  50
CC       TO A SPECIFIED STABILITY VALUE AND GENERATE AN ARRAY OF X,Y     SEAR  60
CC       POINTS DEFINING A CONTOUR OF THE SPECIFIED STABILITY.           SEAR  70
C                                                                        SEAR  80
```

```
C     SAMPLE OF THE CALLING ARGUMENT                                          SEAR  90
C                                                                             SEAR 100
C        CALL SEARCH(NCASE,X,Y,NDIM)                                          SEAR 110
C                                                                             SEAR 120
C     DESCRIPTION OF PARAMETERS                                               SEAR 130
C                                                                             SEAR 140
C        NCASE - DEFINES THE INSTABILITY CASE (INCIPIENT,CRITICAL OR          SEAR 150
C                FULLY DEVELOPED) TO BE USED WHEN GENERATING THE X,Y          SEAR 160
C                ARRAYS.                                                      SEAR 170
C                    NCASE = -1   INCIPIENT INSTABILITY CRITERION             SEAR 180
C                    NCASE =  0   CRITICAL INSTABILITY CRITERION              SEAR 190
C                    NCASE =  1   FULLY DEVELOPED INSTABILITY CRITERION       SEAR 200
C                                                                             SEAR 210
C        X     - THE ARRAY OF ALPHA REAL COORDINATES DEFINING THE             SEAR 220
C                LOCATION OF THE POINTS OF SPECIFIED INSTABILITY.             SEAR 230
C                                                                             SEAR 240
C        Y     - THE ARRAY OF ALPHA IMAGINARY COORDINATES DEFINING THE        SEAR 250
C                LOCATION OF THE POINTS OF THE SPECIFIED INSTABILITY.         SEAR 260
C                                                                             SEAR 270
C        NDIM - THE ORDER OF THE MAP ARRAY.                                   SEAR 280
C                                                                             SEAR 290
C     OTHER ROUTINES NEEDED                                                   SEAR 300
C                                                                             SEAR 310
C        STATEMENT FUNCTION CRIT AND SUBROUTINE INTERP.                       SEAR 320
C                                                                             SEAR 330
C.....                                                                        SEAR 340
      SUBROUTINE SEARCH (NCASE,X,Y,K,NDIM)                                    SEAR 350
      DIMENSION X(500), Y(500)                                                SEAR 360
      COMMON /ARAY/ G(41,41),AR(41),AI(41)                                    SEAR 370
C                                                                             SEAR 380
CC    DEFINE THE STATEMENT FUNCTION CRIT(NCASE,ALPHA)                         SEAR 390
C                                                                             SEAR 400
      CRIT(NCASE,ALPHA) = FLOAT(NCASE)*ABS(ALPHA)                             SEAR 410
C                                                                             SEAR 420
C                                                                             SEAR 430
      K = 0                                                                   SEAR 440
      MDIM = NDIM-1                                                           SEAR 450
C                                                                             SEAR 460
CC    SEARCH FOR SIGN CHANGES BY COLUMN & INTERPOLATE FOR                     SEAR 470
CC    ALPHA IMAGINARY AT WHICH SIGN CHANGE OCCURS                             SEAR 480
C                                                                             SEAR 490
C                                                                             SEAR 500
      DO 5 I=1,NDIM                                                           SEAR 510
C                                                                             SEAR 520
      DO 5 J=1,MDIM                                                           SEAR 530
      IF (G(I,J)-CRIT(NCASE,AR(I))) 2,4,1                                     SEAR 540
    1 IF (G(I,J+1)-CRIT(NCASE,AR(I))) 3,3,5                                   SEAR 550
    2 IF (G(I,J+1)-CRIT(NCASE,AR(I))) 5,3,3                                   SEAR 560
```

```
    3 Y1 = G(I,J)-CRIT(NCASE,AR(I))                                          SEAR 570
      Y2 = G(I,J+1)-CRIT(NCASE,AR(I))                                        SEAR 580
      CALL INTERP (AI(J),AI(J+1),Y1,Y2,AIVAL)                                SEAR 590
      K = K+1                                                                SEAR 600
      X(K) = AR(I)                                                           SEAR 610
      Y(K) = AIVAL                                                           SEAR 620
      GO TO 5                                                                SEAR 630
    4 K = K+1                                                                SEAR 640
      X(K) = AR(I)                                                           SEAR 650
      Y(K) = AI(J)                                                           SEAR 660
    5 CONTINUE                                                               SEAR 670
C                                                                            SEAR 680
CCC                                                                          SEAR 690
CCC   SEARCH FOR SIGN CHANGE BY ROWS AND INTERPOLATE FOR ALPHA               SEAR 700
CCC   REAL AT WHICH SIGN CHANGE OCCURS                                       SEAR 710
CCC                                                                          SEAR 720
C                                                                            SEAR 730
      DO 10 I=1,NDIM                                                         SEAR 740
C                                                                            SEAR 750
      DO 10 J=1,MDIM                                                         SEAR 760
      IF (G(J,I)-CRIT(NCASE,AR(J))) 7,9,6                                    SEAR 770
    6 IF (G(J+1,I)-CRIT(NCASE,AR(J+1))) 8,8,10                               SEAR 780
    7 IF (G(J+1,I)-CRIT(NCASE,AR(J+1))) 10,8,8                               SEAR 790
    8 Y1 = G(J,I)-CRIT(NCASE,AR(J))                                          SEAR 800
      Y2 = G(J+1,I)-CRIT(NCASE,AR(J+1))                                      SEAR 810
      CALL INTERP (AR(J),AR(J+1),Y1,Y2,ARVAL)                                SEAR 820
      K = K+1                                                                SEAR 830
      X(K) = ARVAL                                                           SEAR 840
      Y(K) = AI(I)                                                           SEAR 850
      GO TO 10                                                               SEAR 860
    9 K = K+1                                                                SEAR 870
      X(K) = AR(J)                                                           SEAR 880
      Y(K) = AI(I)                                                           SEAR 890
   10 CONTINUE                                                               SEAR 900
C                                                                            SEAR 910
      RETURN                                                                 SEAR 920
      END                                                                    SEAR 930
C
CCC.....SUBROUTINE INTERP(X1,X2,Y1,Y2,X3).....INTE  10
CCC                                            INTE  20
CCC   PURPOSE                                  INTE  30
CCC                                            INTE  40
CCC      TO LINEARLY INTERPOLATE FOR THE POINT OF ACTUAL SIGN  INTE  50
CCC      CHANGE (X3) BETWEEN TWO POINTS (Y1 & Y2) OF OPPOSITE  INTE  60
CCC      SIGN. THE X-COORDINATES OF Y1 & Y2 ARE X1 & X2, RESPECTIVELY.  INTE  70
CCC                                            INTE  80
C     SAMPLE OF THE CALLING ARGUMENT           INTE  90
```

```
C                                                                       INTE  100
C        CALL INTERP(X1,X2,Y1,Y2,X3)                                    INTE  110
C                                                                       INTE  120
C     DESCRIPTION OF PARAMETERS                                         INTE  130
C                                                                       INTE  140
C        X1 & X2 - X-COORDINATES OF POINTS Y1 & Y2 RESPECTIVELY.        INTE  150
C                                                                       INTE  160
C        Y1 & Y2 - TWO POINTS OF OPPOSITE SIGN FOR WHICH THE POINT      INTE  170
C                  OF ACTUAL SIGN CHANGE (Y = 0) IS TO BE INTERPOLATED. INTE  180
C                                                                       INTE  190
C        X3      - THE VALUE OF X FOR WHICH Y = 0.                      INTE  200
C                                                                       INTE  210
C     OTHER ROUTINES NEEDED                                             INTE  220
C                                                                       INTE  230
C        NONE                                                           INTE  240
C                                                                       INTE  250
C.......................................................................INTE  260
C                                                                       INTE  270
      SUBROUTINE INTERP (X1,X2,Y1,Y2,X3)                                INTE  280
      X3 = (X2*Y1-X1*Y2)/(Y1-Y2)                                        INTE  290
      RETURN                                                            INTE  300
      END                                                               INTE  310
C
C.....................................BLOCK DATA.......................BLKD   10
C                                                                       BLKD   20
C     PURPOSE                                                           BLKD   30
C                                                                       BLKD   40
C        TO INITIALIZE COMMON ARRAYS G1,AR1 & AI1 TO ZERO.              BLKD   50
C                                                                       BLKD   60
C     SAMPLE OF CALLING ARGUMENT                                        BLKD   70
C                                                                       BLKD   80
C        NONE                                                           BLKD   90
C                                                                       BLKD  100
C     DESCRIPTION OF PARAMETERS                                         BLKD  110
C                                                                       BLKD  120
C        G1  - THE MAP OF STABILITY VALUES GENERATED BY PIPEQ.          BLKD  130
C              EACH ELEMENT OF G1 IS A VALUE OF GAMMA* CORRESPONDING    BLKD  140
C              TO A SPECFIC VALUE OF THE REAL AND IMAGINARY PARTS OF    BLKD  150
C              THE WAVE NUMBER, ALPHA.                                  BLKD  160
C                                                                       BLKD  170
C        AR1 - THE LINEAR ARRAY OF X-COORDINATES OF THE STABILITY MAP   BLKD  180
C              (THE REAL PART OF THE WAVE NUMBER, ALPHA).               BLKD  190
C                                                                       BLKD  200
C        AI1 - THE LINEAR ARRAY OF Y-COORDINATES OF THE STABILITY MAP   BLKD  210
C              (THE IMAGINARY PART OF THE WAVE NUMBER, ALPHA).          BLKD  220
C                                                                       BLKD  230
C     OTHER ROUTINES NEEDED                                             BLKD  240
```

```
C                                                                       BLKD 250
C         NONE                                                          BLKD 260
CCCCC                                                                   BLKD 270
CCCCC.....................................................................BLKD 280
C                                                                       BLKD 290
      BLOCK DATA                                                        BLKD 300
      COMMON /ARAY/ G1(41,41),AR1(41),AI1(41)                           BLKD 310
      DATA G1,AR1,AI1/1681*0.0,82*0.0/                                  BLKD 320
      END                                                               BLKD 330
C
CCCCC.....SUBROUTINE CHART(SN,SREY,SLAMDA).....................CHAR  10
CCCCC                                                                   CHAR  20
CCCCC PURPOSE                                                           CHAR  30
CCCCC                                                                   CHAR  40
CCCCC    TO LABEL THE CONTOUR PLOT                                      CHAR  50
CCCCC                                                                   CHAR  60
CCCCC SAMPLE OF THE CALLING ARGUMENT                                    CHAR  70
CCCCC                                                                   CHAR  80
CCCCC    CALL CHART(SN,SREY,SLAMDA)                                     CHAR  90
CCCCC                                                                   CHAR 100
CCCCC DESCRIPTION OF PARAMETERS                                         CHAR 110
CCCCC                                                                   CHAR 120
CCCCC    SN      - THE NUMBER OF INTERIOR MESH POINTS USED  FOR THE     CHAR 130
CCCCC              STABILITY CONTOUR MAP BEING PLOTTED.                 CHAR 140
CCCCC                                                                   CHAR 150
CCCCC    SREY    - REYNOLDS NUMBER                                      CHAR 160
CCCCC                                                                   CHAR 170
CCCCC    SLAMDA - THE NONUNIFORM MESH PARAMETER APPLICABLE TO THE       CHAR 180
CCCCC             DATA BEING PLOTTED.                                   CHAR 190
CCCCC                                                                   CHAR 200
CCCCC OTHER ROUTINES NEEDED                                             CHAR 210
CCCCC                                                                   CHAR 220
CCCCC    ONLY BUILT-IN VERSATEC PLOTTING FUNCTIONS NEWPEN,SYMBOL &      CHAR 230
CCCCC    NUMBER.  NOTE THAT THESE ROUTINES MAY ONLY BE ACCESSED WHEN    CHAR 240
CCCCC    RUNNING UNDER 'FORTCLGW'.                                      CHAR 250
CCCCC                                                                   CHAR 260
CCCCC.....................................................................CHAR 270
C                                                                       CHAR 280
      SUBROUTINE CHART (SN,SREY,SLAMDA)                                 CHAR 290
C                                                                       CHAR 300
CC                                                                      CHAR 310
      XO = 2.5                                                          CHAR 320
      YO = 6.5                                                          CHAR 330
      HT = 0.15                                                         CHAR 340
      HT1 = 0.7*HT                                                      CHAR 350
      DELY1 = .08+HT                                                    CHAR 360
      DELY2 = .065+HT1                                                  CHAR 370
```

```
      DELX = .1                                                          CHAR 380
C                                                                        CHAR 390
CC    GRAPH TITLE                                                        CHAR 400
C                                                                        CHAR 410
      CALL NEWPEN (2)                                                    CHAR 420
      CALL SYMBOL(XO,YO,HT,'STABILITY CONTOUR PLOT',0.,22)               CHAR 430
      XO = XO+3.*DELX                                                    CHAR 440
      YC = YO-DELY1                                                      CHAR 450
      CALL SYMBOL(XO,YO,HT,'FOR THE CASE N = 0',0.,18)                   CHAR 460
C                                                                        CHAR 470
CC    MESH VALUE                                                         CHAR 480
C                                                                        CHAR 490
      CALL NEWPEN (1)                                                    CHAR 500
      XO = XO+4.*DELX                                                    CHAR 510
      YO = YO-DELY1                                                      CHAR 520
      CALL SYMBOL(XO,YO,HT1,'NMESH  = ',0.,9)                            CHAR 530
      CALL NUMBER (999.,999.,HT1,SN,0.,-1)                               CHAR 540
C                                                                        CHAR 550
CC    REY VALUE                                                          CHAR 560
C                                                                        CHAR 570
      YO = YO-DELY2                                                      CHAR 580
      CALL SYMBOL(XO,YO,HT1,'REY    = ',0.,9)                            CHAR 590
      CALL NUMBER (999.,999.,HT1,SREY,0.,-1)                             CHAR 600
C                                                                        CHAR 610
CCC   LAMBDA VALUE                                                       CHAR 620
C                                                                        CHAR 630
      YC = YO-DELY2                                                      CHAR 640
      CALL SYMBOL(XO,YO,HT1,'LAMBDA = ',0.,9)                            CHAR 650
      CALL NUMBER (999.,999.,HT1,SLAMDA,0.,1)                            CHAR 660
C                                                                        CHAR 670
CC    STABILITY AREA LABELS                                              CHAR 680
CCCCC NOTE - SINCE THE SHAPE OF THE CURVE VARIES WITH                    CHAR 690
CCCCC        EACH SET OF INPUT DATA, THE COORDINATES OF THE FOLLOWING    CHAR 700
CCCCC        LABELS MUST BE ADJUSTED FOR EACH SPECIFIC PLOT.             CHAR 710
C                                                                        CHAR 720
      CALL NEWPEN (2)                                                    CHAR 730
      CALL SYMBOL(4.0,4.5,HT1,'SUPERCRITICAL',0.,13)                     CHAR 740
      CALL SYMBOL(5.6,4.5,HT1,'SUBCRITICAL',0.,11)                       CHAR 750
      CALL SYMBOL(6.9,4.5,HT1,'STABLE',0.,6)                             CHAR 760
      YC = 4.5-DELY2                                                     CHAR 770
      CALL SYMBOL(4.0,YO,HT1,' INSTABILITY',0.,12)                       CHAR 780
      CALL SYMBOL(5.6,YO,HT1,'INSTABILITY',0.,11)                        CHAR 790
C                                                                        CHAR 800
      RETURN                                                             CHAR 810
      END                                                                CHAR 820
C                                                                        CHAR 830
CC
```

```
C.....
C
C     THE FOLLOWING CARDS COMPRISE THE DATA DECK FOR PROGRAM STBCONT.
C
C        /*
C        //GO.SYSIN DD *
C                 .
C                 .
C        DATA DECK FROM ONE RUN OF PROGRAM PIPEO (MODENO = 2)
C                 .
C                 .
C        /*
C
C.....
```

LIST OF REFERENCES

1. Davey, A., and Drazin, P.G., "The Stability of Poiseuille Flow in a Pipe," Journal of Fluid Mechanics, v. 36, part 2, p. 209, 22 August 1968.

2. Garg, V.K., and Rouleau, W.T., "Linear Spatial Stability of Pipe Poiseuille Flow," Journal of Fluid Mechanics, v. 54, part 1, p. 113, 6 January 1969.

3. Gill, A.E., "The Least-Damped Disturbance to Poiseuille Flow in a Circular Pipe," Journal of Fluid Mechanics, v. 61, part 1, p. 765, 3 December 1973.

4. Harrison, W.F., On the Stability of Poiseuille Flow, Ae. E. Thesis, Naval Postgraduate School, Monterey, California, 1975.

5. Huang, L.M. and Chen, T.S., "Stability of Developing Flow Subject to Non-axisymmetric Disturbances," Journal of Fluid Mechanics, v. 63, part 1, p. 183, 16 April 1973.

6. Johnston, R.H. III, A Program for the Stability Analysis of Pipe Poiseuille Flow, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1976.

7. Leite, R.J., An Experimental Investigation of Axially Symmetric Poiseuille Flow, Report No. OSR-TR-56-2, Air Force Contract AF18(600)-350, November, 1956.

8. Naval Postgraduate School Report NPS-67Gn77051, Improved Finite Difference Formulas for Boundary Value Problems, by T.H. Gawain and R.E. Ball, 1 May 1977.

9. Naval Postgraduate School Report NPS67-78-006, A Basic Reformulation of the Pipe Flow Stability Problem and Some Preliminary Numerical Results, by T.H. Gawain, 1 September 1978.

10. Reynolds, O., "An Experimental Investigation of the Circumstances which Determine whether the Motion of Water Shall be Direct or Sinuous, and the Law of Resistance in Parallel Channels," Phil. Trans. Royal Soc., 174, p. 935-982, 1883.

11. Salwen, H., and Grosch, C.E., "The Stability of Poiseuille Flow in a Pipe of Circular Cross-section," Journal of Fluid Mechanics, v. 54, part 1, p. 93, 6 March 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Prof. T.H. Gawain, Code 67Gn Department of Aeronautics Naval Postgraduate School Monterey, California 93940	5
5.	LT Michael James Arnold, USN 10825 Single Tree Lane Spring Valley, California 92077	1